HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

Delivery Automation Tool for ISAVIA
Project Report (Verkefnislýsing)

T-404-LOKA

Students:
Árni Þorvaldsson (arnitho12@ru.is)
Freyr Bergsteinsson (freyrb12@ru.is)
Gunnar Þór Helgason (gunnarh07@ru.is)
Sigurbjörn Kristjánsson (sigurbjorn12@ru.is)

Instructor:
Daníel Máni Jónsson (daniel@valitor.is)

Examiner:
Hannes Pétursson (hap@ru.is)

Table of Contents

Document History

Date	Author	Description
2015/01/20	ÁÞ, FB, GÞH, SK	Initial version.
2015/02/12	FB	Updated overview and added Internals chapter
2015/03/07	GÞH	Updated estimate time to finish project
2015/03/10	ÁÞ	Updated master script diagram and minor revisions

Introduction

The company

Tern Systems is an Icelandic company that reaches back to the late 1970's, but was formally established in 1997 by the University of Iceland and the Icelandic Aviation Administration. Tern Systems makes software for air-traffic control that is being used all around the world. Tern Systems is also maintaining and developing software for Isavia, its parent company. Isavia handles the operation and development of all airports in Iceland and manages air traffic in the Icelandic control area.

The official language in Tern Systems is English and thus all reports and documents regarding the project will be in English.

The product

Software delivery for software maintained by Isavia often proves to be a tedious task sometimes requiring a developer to spend several hours manually checking to see if everything is in order. The product created by this project is called the Delivery Automation Tool and it aims to solve that problem by automating the delivery process. For clarification, "the product" is what is being developed for Tern Systems, while "the project" is the final year project at Reykjavík University.

The team

Work on the project will be done in compliance with Agile methodology. The team that will work on the project has been given the name *Autobots* and its members, along with their respective Scrum roles are:

Árni Þorvaldsson <arnitho12@ru.is> - Developer
Freyr Bergsteinsson <freyrb12@ru.is> - Scrum Master, Developer
Gunnar Þór Helgason <gunnarh07@ru.is> - Developer
Sigurbjörn Kristjánsson <sigurbjorn12@ru.is> - Developer

The duration of the first Scrum sprint, Sprint Zero, was 7 days from January 19th to January 26th beginning and ending on a Monday.

The succeeding sprints will have a duration of 14 days, also beginning on and ending on Mondays, with sprint review (including a demo), retrospective analysis and planning during the meetings. The lengths of the sprints will be discussed in sprint reviews and might change.

At the start of the project Trello was being used to keep track of the Scrum board and backlog, but it was soon replaced with Visual Studio Online, which is better suited to the requirements of the team. A journal is shared on Google Drive to keep track of the group members working hours.

Team meetings are usually held at Tern Systems' premises at least three times a week, for about 4-6 hours at a time. Regular meeting will be on-site on Mondays from 12:30-17:30 and on Tuesdays from 19:30-23:30. Other meeting times will be scheduled on a per-week basis and will be either on-site or off-site depending on what's being worked on.

Daily meetings will be held at the start of the regular meetings on Mondays and Tuesdays. On Wednesdays, Thursdays and Fridays the team will have daily meetings through Google Hangout at 14:00. No daily meetings are scheduled for Saturdays or Sundays.

Communication with company

The team's correspondent at Tern Systems is Heiðar Harðarson <heidar@tern.is>. He holds the role of Product Owner for this project. Mr. Harðarson has been employed by Tern Systems since 2000 as a programmer and project manager. He leads the team at Tern Systems that is responsible for development and maintenance of systems owned by Isavia. This gives him a very clear vision in regards to what the end-product of this project should feature.

The team meets the correspondent at Tern Systems weekly, on Mondays at 13:00, and presents the products of the week.

Computers are available for loan from Tern Systems, to be used at the premises and returned at the completion of the project. Each group member can choose his development environment according to his own preferences. Text editors are used for coding, as the project is made mostly of Python files, as well as HTML/CSS/JS for the web-based front end.

While the project is on-going, all code will be kept on a private repository on github. Thus, it is possible to work on the project outside of the workplace, and also to show the project's progress, as expected in status meetings 1 and 3. After completion of the project, the code will be handed to Tern Systems and the github repository deleted.

Tern is the owner of the product and holds copyrights.

End product

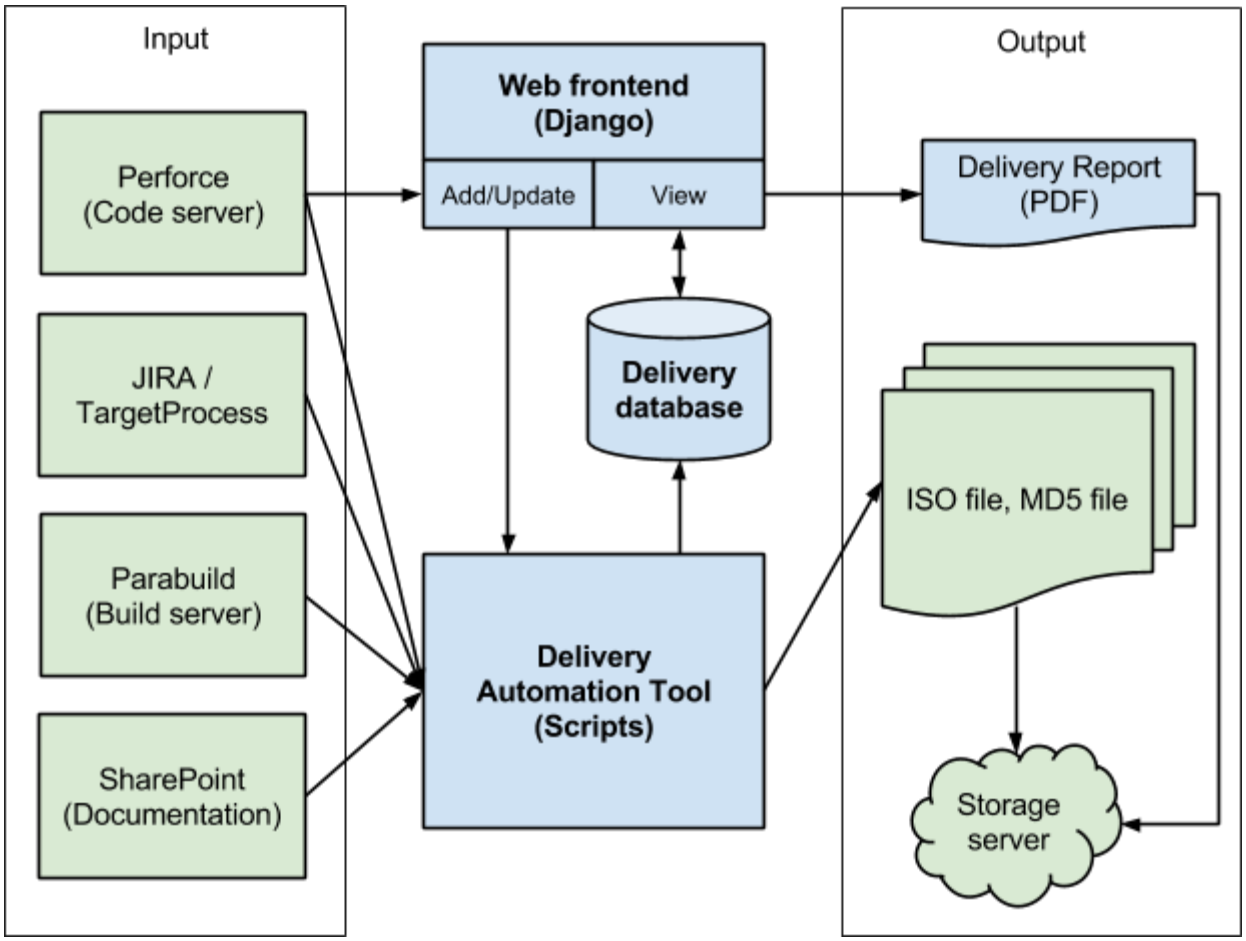
In addition to the software developed for Tern Systems, the end product for the project itself will include the following reports:

- Project Report (Verkefnislýsing) - This report
- Working Arrangement (Verkskipulag)
- Project Schedule (Verkáætlun)
- Risk Analysis (Áhættugreining)
- Developments Report (Framvinduyfirlit)
- Operational Manual (Rekstrarhandbók)
- User Guide (Notendaleiðbeiningar)
- Product Backlog (Kröfulýsingar)
- Design Document (Hönnunarskýrsla)
- Work Journal (Vinnudagbók)

Disclaimer: Some of these reports may be merged.

Overview and system architecture

The Delivery Automation Tool interacts with several other systems that are operated on the intranet at Tern Systems. It collects information for those systems and generates the desired output. The following image shows the overview of the system architecture where objects in blue is software that is created for this project, while objects in green is software that is already used by Tern Systems.



Input

The Delivery Automation Tool and its front end uses data from 4 different systems:

- **Perforce** is a revision control system. It keeps track of all the code created at Tern Systems. The web front end uses perforce to populate which products and versions are available for delivery. Moreover, it is the job of the Delivery Automation Tool to use

Perforce to verify that the delivery contains the correct information.

- **Parabuild** is a continuous integration service. When a product is being delivered it is a prerequisite that is has been successfully built by the build server. The Delivery Automation Tools' job is to retrieve the correct version of the desired product to delivery.

- **JIRA** is a issue tracking tool. When a product is being delivered it is a prerequisite that all Isavia tasks pertaining to a version of a product have been created in JIRA and are marked as completed. **TargetProcess** is a similar product and it is Tern System's intention to eventually replace JIRA with TargetProcess. The design of the Delivery Automation Tool will take this into account.

- **SharePoint** is a framework that is being used by Tern Systems for content management. The Delivery Automation Tool will retrieve documents pertaining to the product being delivered.

Output

The output of the Delivery Automation Tool is the same as when a delivery is manually created.

- An **ISO file** containing all data that is to be delivered to Isavia. This typically includes any binaries and/or source files, documentation and a report of new, modified and removed files since the previous version of the product being delivered.

- An **MD5 file** containing the MD5 sum of the ISO file to make sure the ISO file has not been tampered with during delivery.

- A **PDF file** containing the delivery report. The contents of the delivery report is described later.

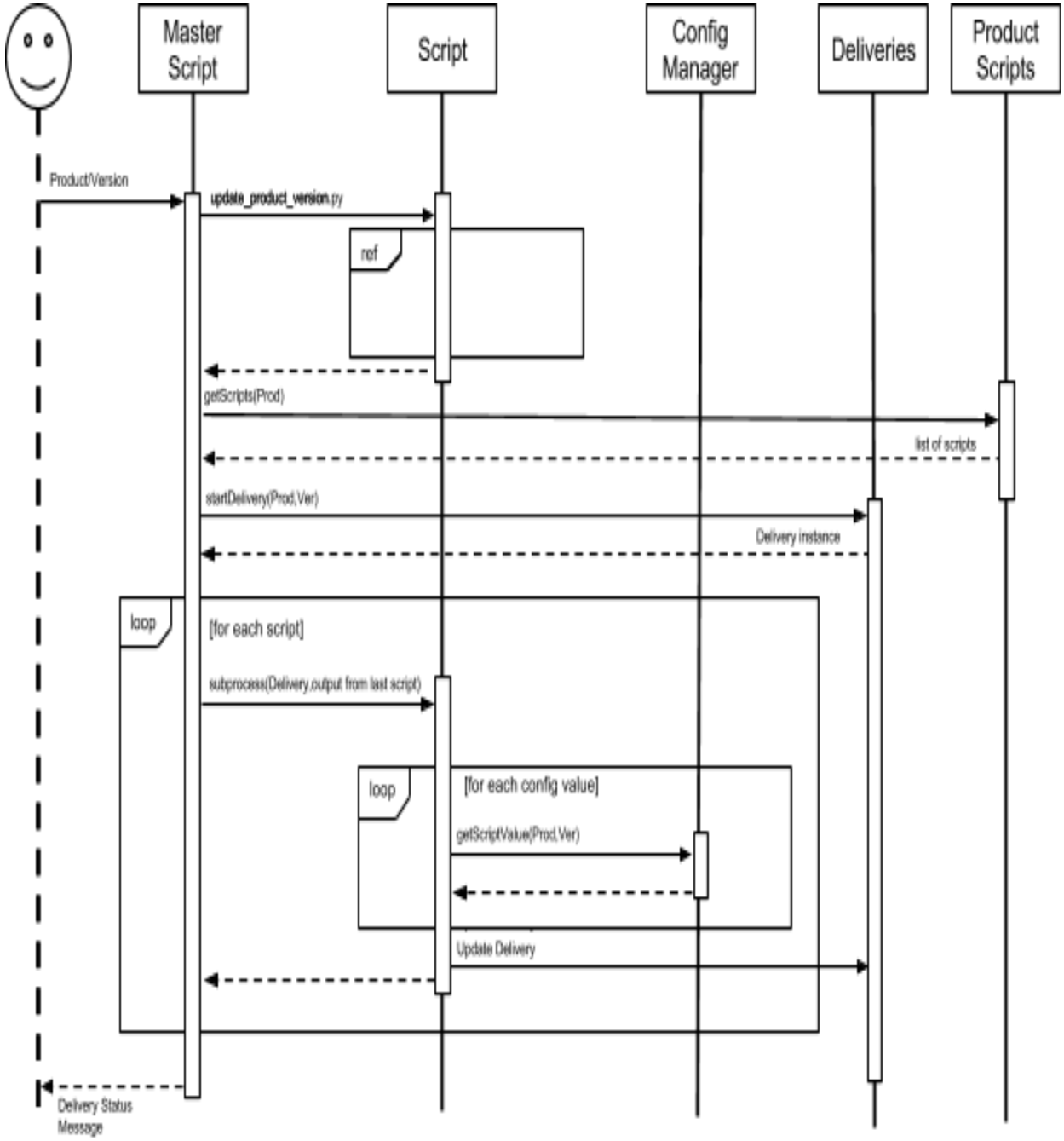
- **Confluence** is a collaboration software and is used by Tern Systems for various internal documentation and tasks. The Delivery Automation Tool will specifically update one page on the Confluence site that lists which deliveries have been made.

Internals of the Delivery Automation Tool

The Delivery Automation Tool relies on a master script, which takes as an input the name and version of the product to create a delivery for. It in turn consults a Config Manager about

which other scripts to execute based on the given product/version, and then goes through executing them, using the output of one script as the input to the next script.

Master Script Diagram



Scope of the Project

Estimated time to finish project: 1200-1400 hours.

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

Tern Systems - Design Document (Hönnunarskýrsla)

T-404-LOKA

Students:
Árni Þorvaldsson (arnitho12@ru.is)
Freyr Bergsteinsson (freyrb12@ru.is)
Gunnar Þór Helgason (gunnarh07@ru.is)
Sigurbjörn Kristjánsson (sigurbjorn12@ru.is)

Instructor:
Daníel Máni Jónsson (daniel@valitor.is)

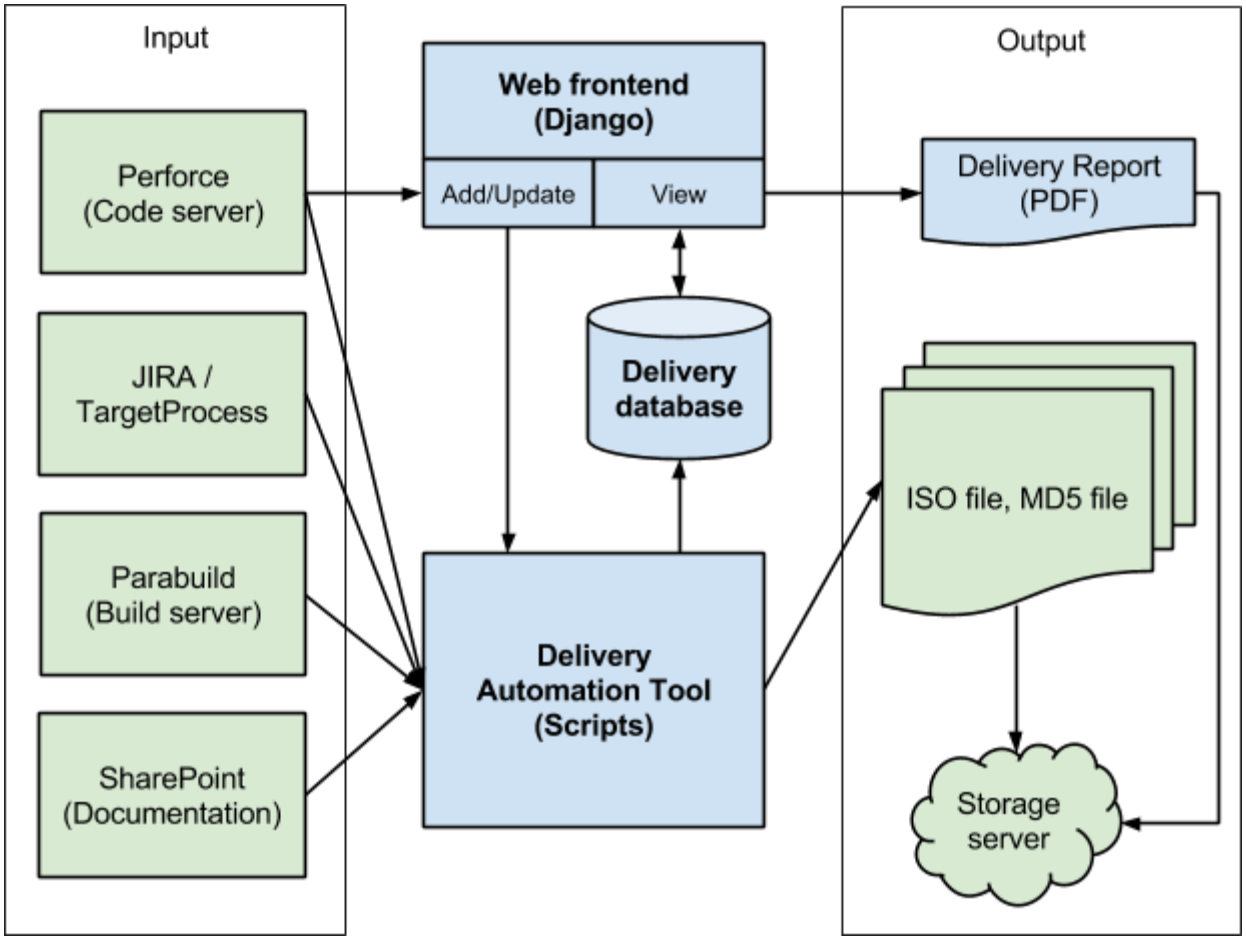
Examiner:
Hannes Pétursson (hap@ru.is)

Table of Contents

Document History

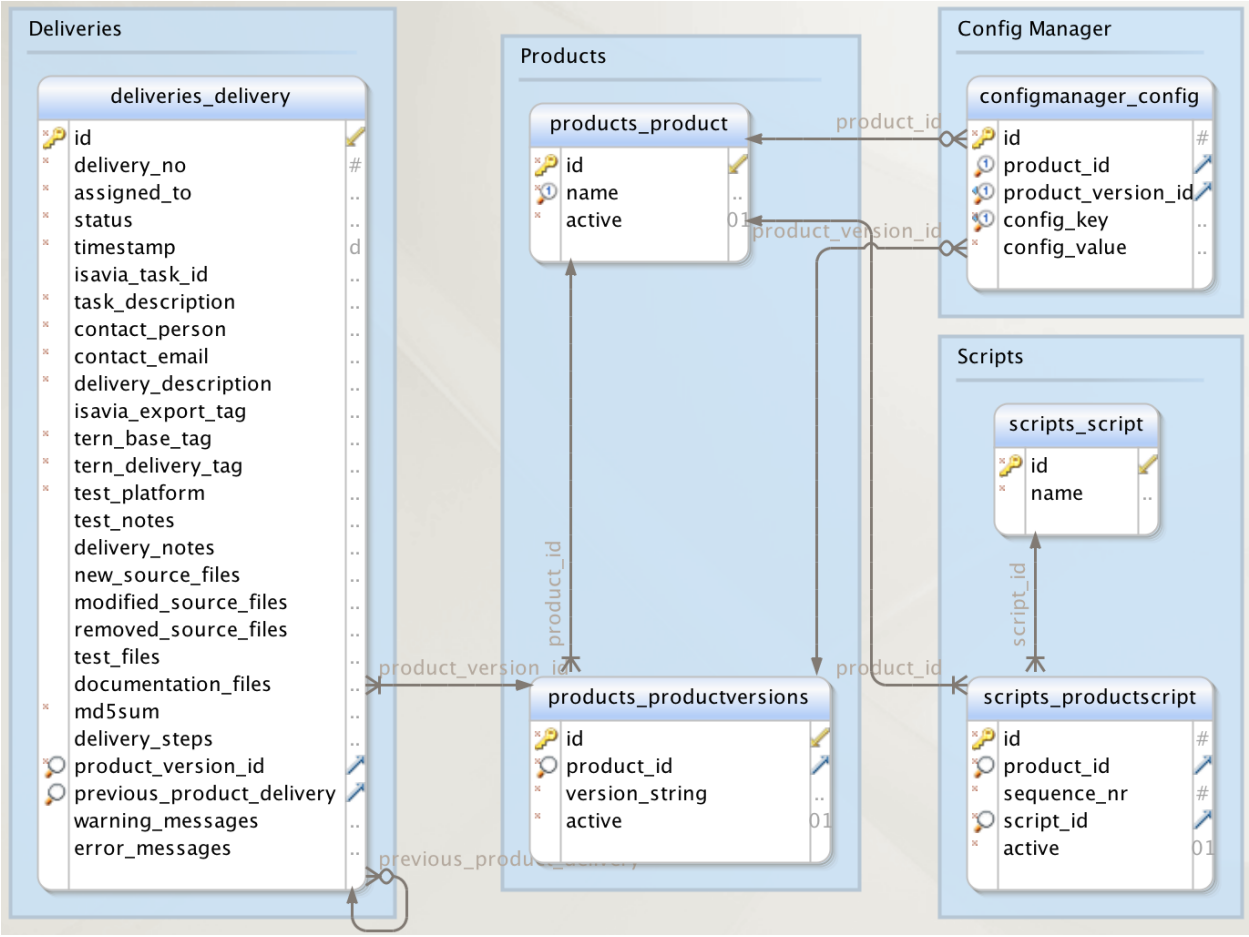
Date	Author	Description
2015-03-04	FB	Initial version.

Overview and system architecture



Database Schema

There are four groups of database tables for this product: Deliveries, Products, Config Manager and Scripts.



Deliveries

The Deliveries group only contains one table, *delivery*. One row in this table represents one delivery from Tern to Isavia, and each field in the table is mapped directly to the fields of the Delivery Report being used, with a few exceptions. The fields *previous_product_delivery*, *warning_messages* and *error_messages* are not fields in the current Delivery Report and are only used for the automation process.

Products

The Products group contains two tables, *products* and *productversions*. The *products* table contains the list of all products that have been and can be delivered. Each product can have one or many versions (also called releases) and which versions have been released for which product is stored in the *productversion* table.

Config Manager

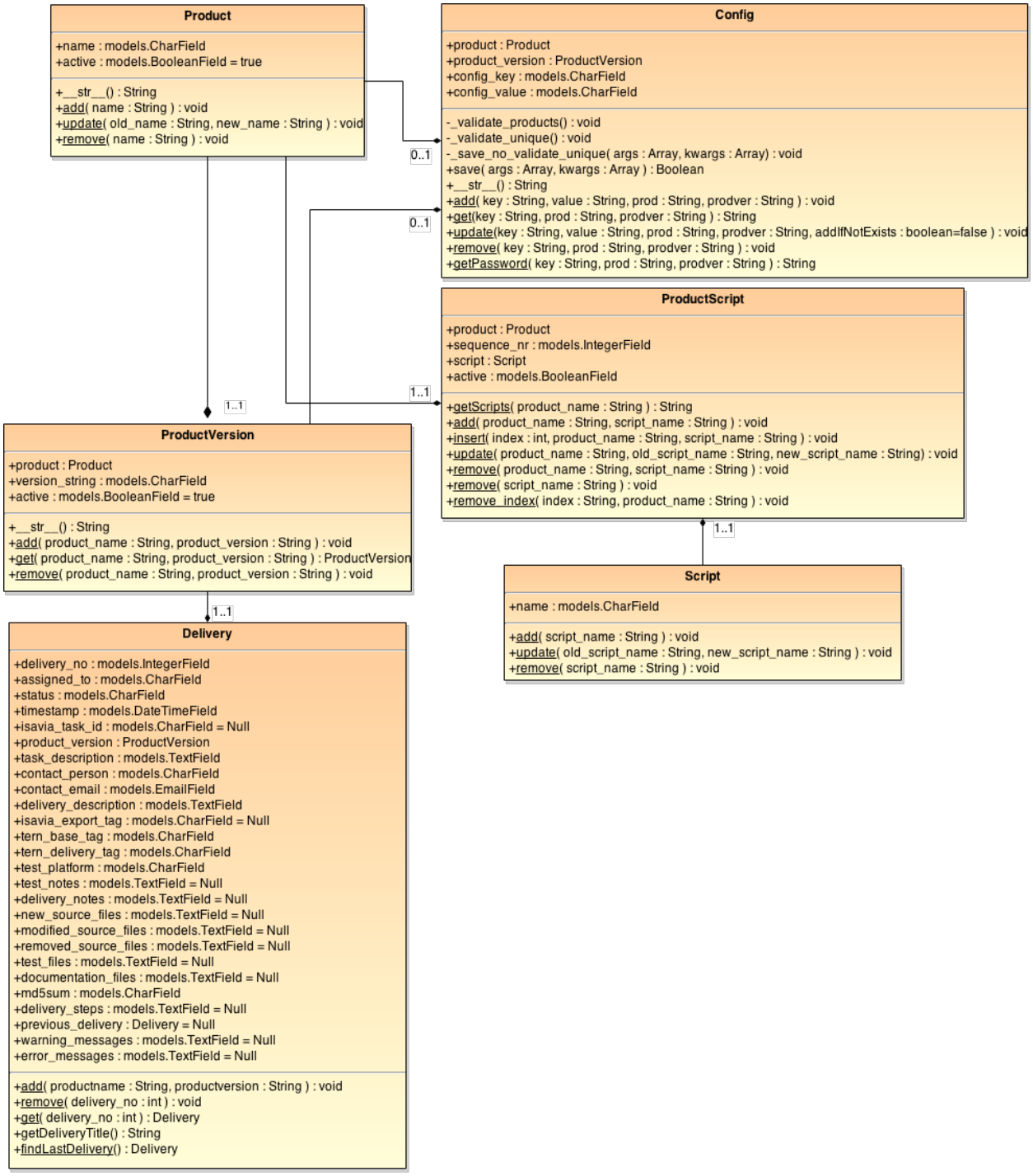
This group only contains one table, *config*. It stores key/value pairs which the automation process uses for configuration. Pairs can be global configuration values, product-specific values or productversion-specific values. If a key/value pair is a global configuration value, then the *product* and *productversion* fields are null. If a key/value pair is product-specific, then the *productversion* field is null, while the *product* field references the product. If a key/value pair is productversion-specific, then the *product* field is null, while the *productversion* field references a productversion. An entry in the config table that has values in both the *product* and the *productversion* fields is an invalid entry.

Scripts

This group is the heart of the automation process. When a delivery is created, then, depending on which product is being released, a series of scripts are run. The *scripts* table contains a list of all scripts available to run, while the *product script* table contains which scripts are executed for which product and in what order.

Django Models

The Django models are direct representations of the underlying database. Each model is a Python class which has a number of properties of types derived from a parent model class. This abstracts the database-specific operations away from the development. Each class has a number of wrapper methods, for handling data validation, insertions, updates and removals.



Credentials

Many of the systems that this product interacts with needs credentials for logging in to their respective sites. The config manager stores these credentials, but the password is not in plaintext. The modules retrieving credentials must use the getPassword() method in the Config Manager to retrieve passwords as they should be used in their respective systems.

The exact implementation of how passwords are stored is subject to change once the product has been delivered to Tern Systems. During development, the passwords are stored with *base64+rot13* encryption simply to have at least some minimal security where the passwords aren't directly readable by a human.

Task Headers

One of the intricacies of deliveries are Task Headers in source files. Task Headers are comments in source files that serve as a commit log of sorts. Task Headers **must** be present in new and modified source files, and they **must** have a correct entry number, date and task number. This information is entered by hand by the developer and will often include errors. There are a few scripts that do some checking on these task headers.

The following is an excerpt from a source file and shows an example of how Task Headers can be represented:

```
/*-----------------------------------------------------------------------------
| Copyright (c) 2009, Tern Systems, Inc.  All Rights Reserved.
|
| 30 29-Jan-2015 Tern Systems/Birgir Ragnarsson
|    Task: 5651
|    Creating an exclusion area for along track deviation reports
|
....
|
| 04 28-april-2009 Flugkerfi/Stefan Palsson
|    Task: 3384
|    Not sending conformance reports if all flight levels below 55
|
| 03 18-jan-2009 Flugkerfi/Stefan Palsson
|    Task: 3328
|    All changes related to the ReportService part of the ISDS project
|
| 02 12-Nov-2008 TERN
|    Task 3276
|    Fixing CR Messages generated outside of BIRD
|
| 01 30-Jan-2006 Christian Johansson
|    Initial version of ICE Services
|
-----------------------------------------------------------------------------*/
```

Note that some task header entries are missing a task number. This is acceptable for legacy code, but unacceptable for new code.

Also note that entry numbers increase by one for each task header entry. Failure to correctly number the task header entries (as can be the case in a hurried copy-paste) is unacceptable.

The date field must have a correct date (not older than the task header entry before it and not newer than the current date) and it must be formatted in the same way as the other task header entries in the same file. Here's an example of task headers from another file in the source repository at Tern:

```
/*-----------------------------------------------------------------------
| Copyright (c) 2005, Tern Systems, Inc.  All Rights Reserved.
|
| 06 2014-12-08 Tern Systems/Birgir Ragnarsson
|    Task: 4915
|    Splitting ICE.xml into functionalConfig.xml and connectionConfig.xml.
|
| 05 2010-10-13 Tern Systems/Ida Kramarczyk
|    Task: 3548
|    Porting Qt3 to Qt4
|
....
```

Note that in this example, dates are in another format, which is acceptable as long as all task header entries in the same file have the same format. Some formats have a yyyy-mm-dd format, others have dd-mm-yyyy. Some formats have numbers for months, others have the name of the month. Some formats have a full name of the month, others have abbreviations.

Some of the scripts specifically check that the task headers are correct and report if there is an error or a warning.

Scripts

Master Script (create_delivery.py)

The master script (whether invoked from a web interface or from command line) runs a series of scripts. Which scripts it runs depends on what is configured in the Scripts module for the given product.

Each script takes as an input:

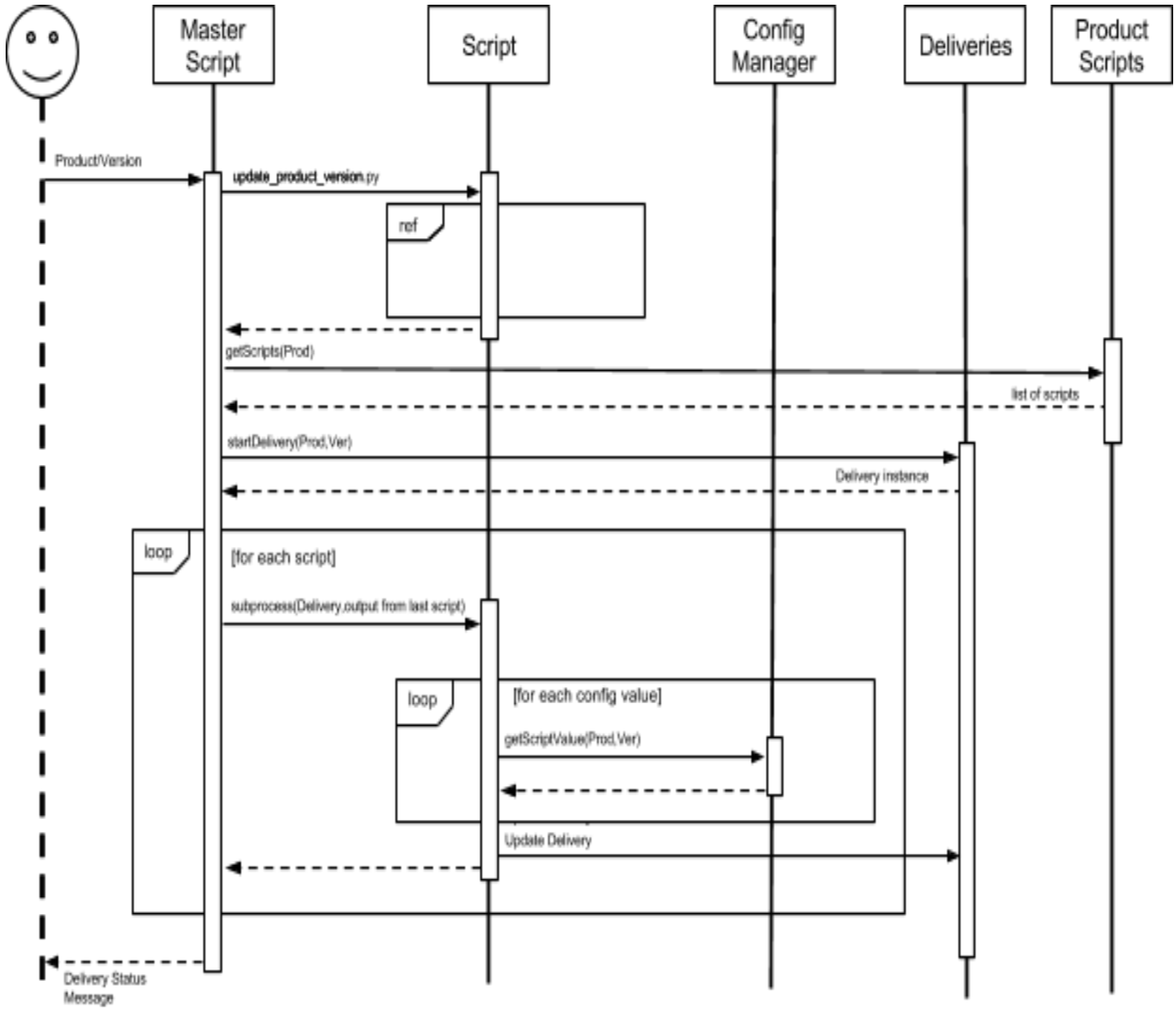
- A Delivery number as an argument

Each script exits with one of 2 possible exit codes:

- 0 - the script completed its task successfully. If there was any incident, then it added/appended text to the *warning_message* field in the delivery instance.
- 1 - the script failed to complete its task. The reason for the failure is written to standard error.

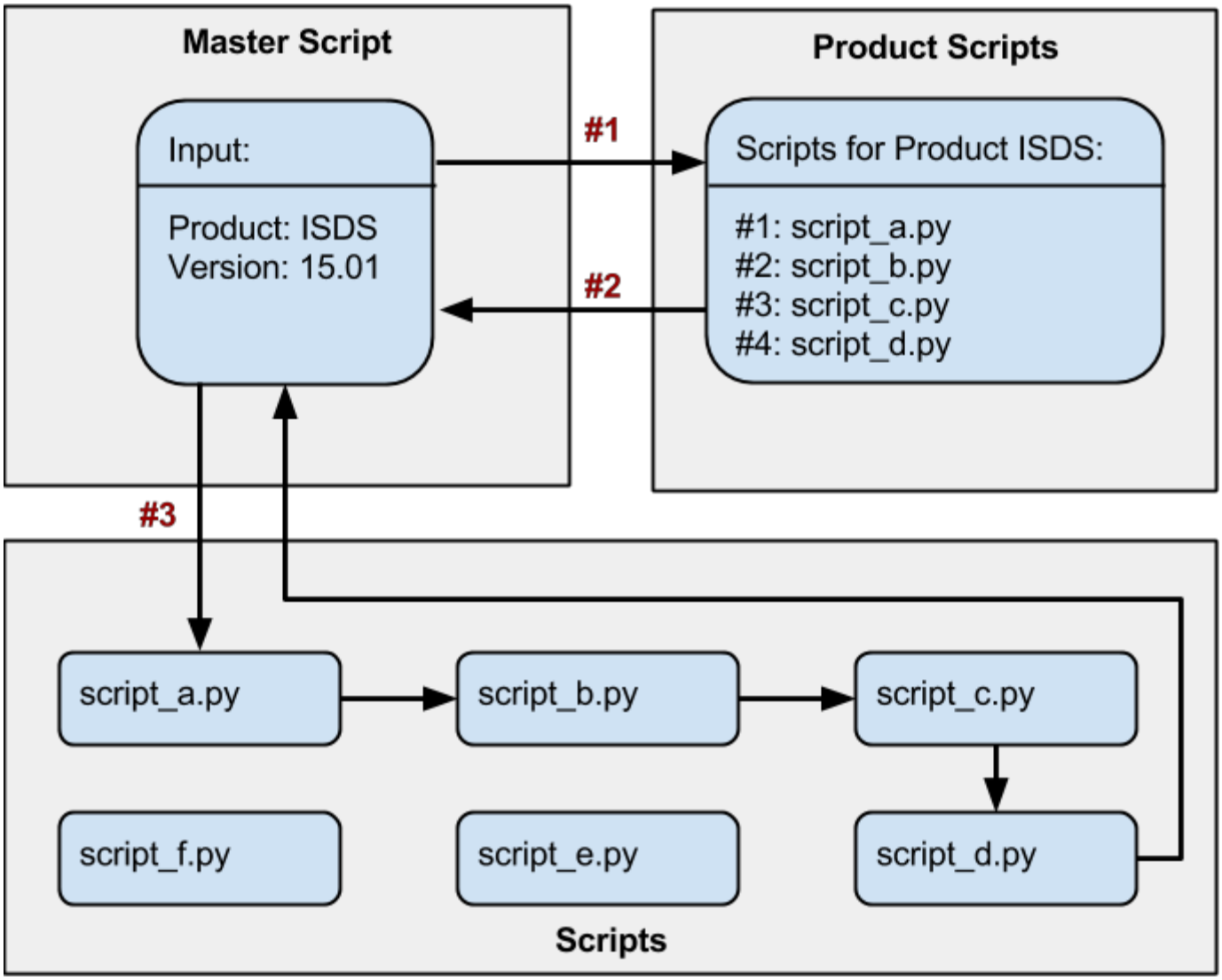
When the last script has finished, then the exit status is checked. If the last script exited normally (exit code 0), then the accumulated *warning_message* is sent to the standard error and the master script also exits with exit code 0. If the last script exited with an error (error code 1), then the master script also exits with exit code 1.

Sequence Diagram



Script Execution Example

The following diagram shows an example of the execution flow for scripts when creating a delivery. In this example, the master script is invoked where the input has a product name *ISDS* and a product version *15.01*. A list of scripts is retrieved from the ProductScript module for the given product, which are then executed one after another, with one script also getting as input the output of the previous script. Not every script available needs to be executed.



Script: update_product_versions.py

Purpose

Check Perforce which products are available for delivery and update the database to reflect that. This script should be called by the web interface before populating which products and versions are available for delivery, as well as at the start by the master script to make sure the data is up-to-date even when run from the command line.

The script retrieves credentials and which path(s) in Perforce to check for products from the Config Manager. An example for a path could be: `//isavia/…`
For each product, check which versions are available for release, using the paths retrieved for the products and for each of them append “/releases/...”. An example for a path that retrieves which versions are available for a product called ISDS could be:
`//isavia/isds/releases/…`

Add products and product versions to the database if they don’t already exist, marking each of them as an active product version. Do not remove products or product versions from the database if they are no longer available, but rather mark them as inactive. These are legacy product and product versions.

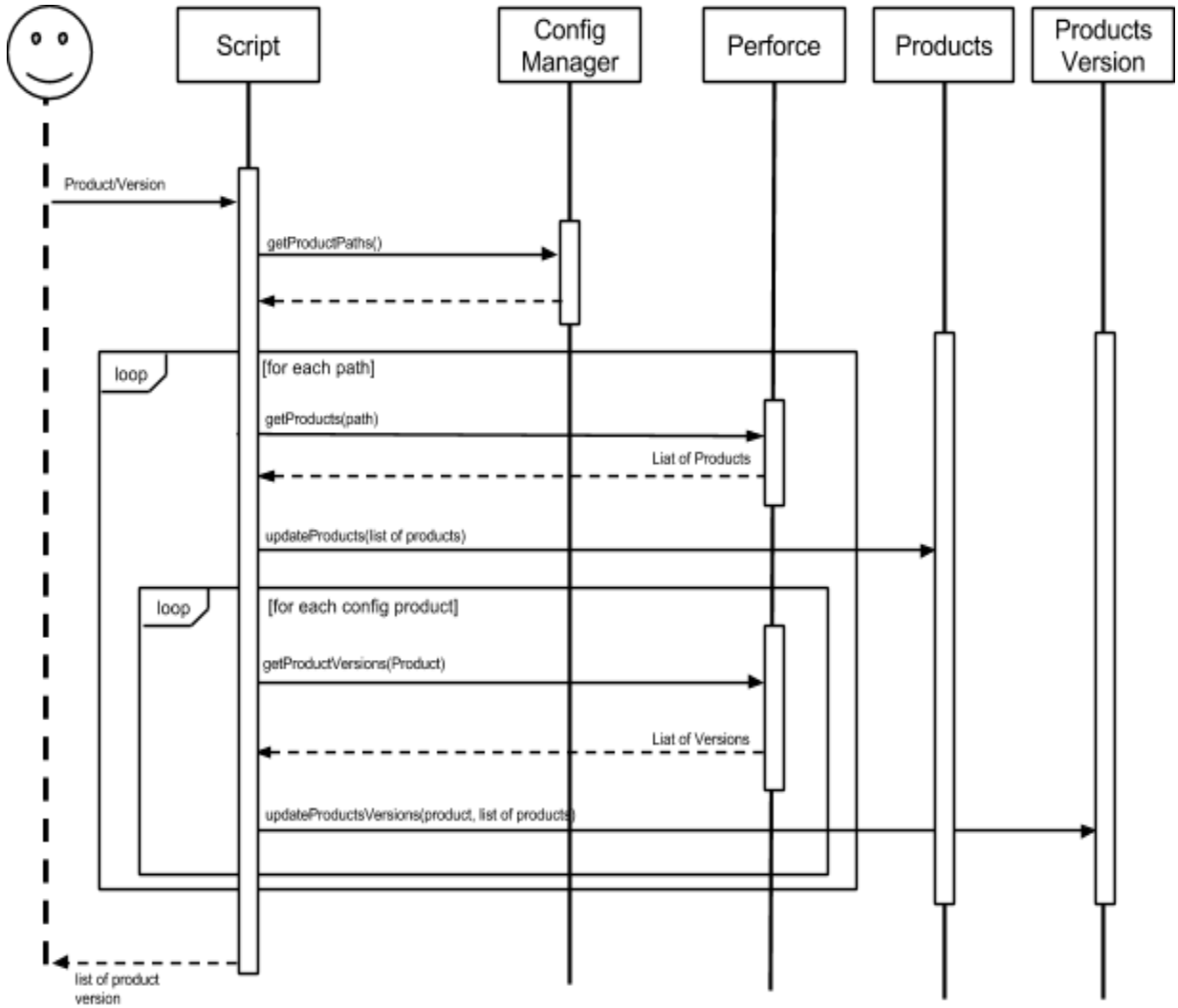
Input

This script does not take any input.

Output

Returns all the active product versions as a string, which each product version separated by a comma. An example for the string returned by this script could be: “ICE 14.02,ICE 14.02.2,ISDS 15.01,ISDS 15.2”

Sequence Diagram



Script: check_task_header_incremental_nr.py

Purpose

This scripts goes through all the files that have been modified since the last delivery for the product currently being delivered (using Perforce diff). For each of those files, it examines the task headers. Each task header entry has an entry number that starts with 1 and increments by 1 for each task header entry. This script makes sure that the task header entries conform to this.

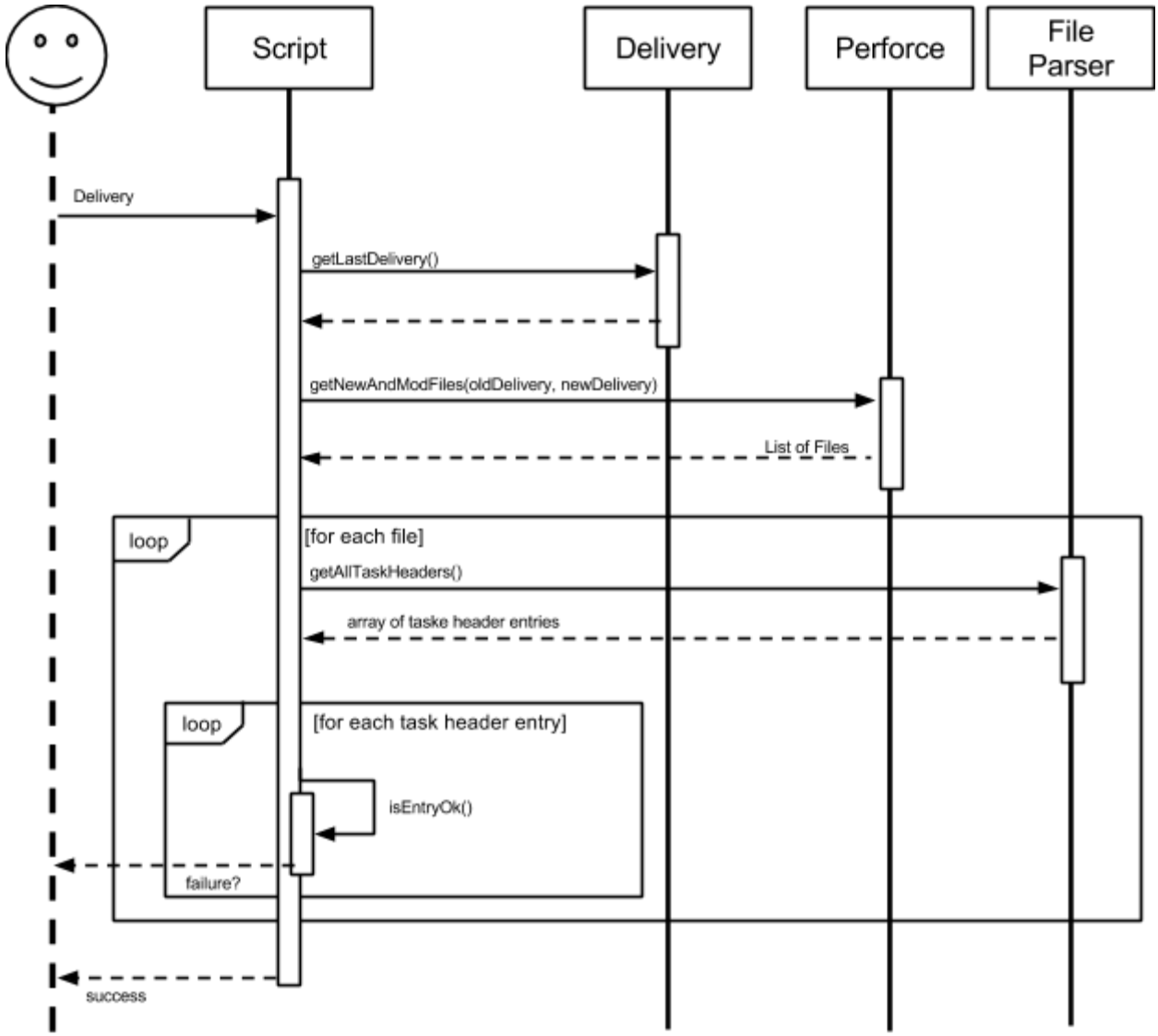
Input

This script takes an instance of a delivery as its input.

Output

If a file's task header list does not conform such that its task header entries do not have its entries numbered correctly, then this script adds an error message to the delivery and exits with error code 1 (this is a fatal error), otherwise it does nothing and exits with error code 0.

Sequence Diagram



Script: check_task_header_dates.py

Purpose

This scripts goes through all the files that have been modified since the last delivery for the product currently being delivered (using Perforce diff). For each of those files, it examines the task headers. This script makes sure that the date format in each task header entry is the same within that file. The script also makes sure that the dates in the task headers that are new since the last delivery are dates that are later than the last delivery date and earlier or equal to the current date.

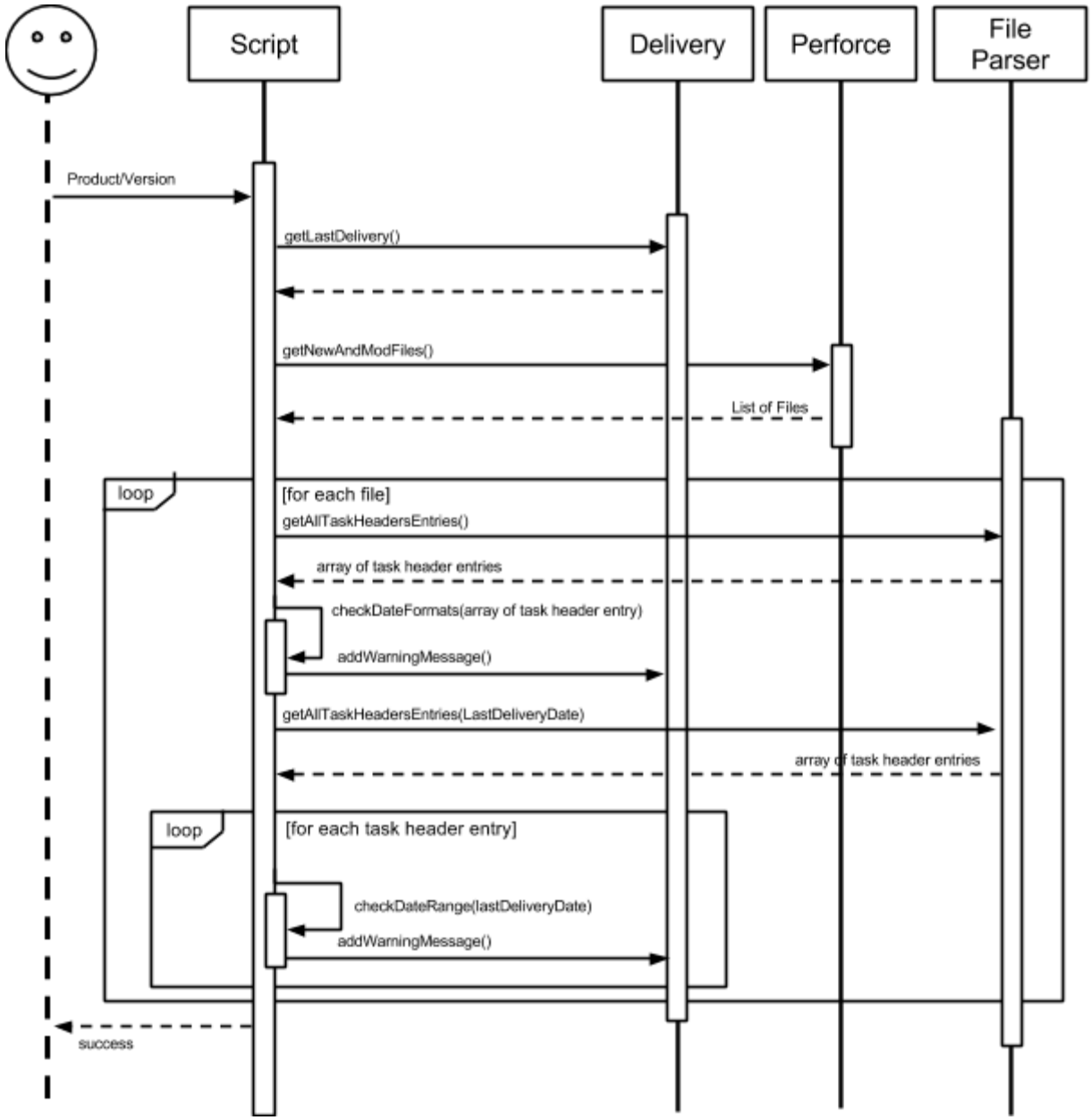
Input

This script takes an instance of a delivery as its input.

Output

If some task header entry is missing a date, or the date is incorrectly formatted compared to the date format of the other task header entries, or if the date is inconsistent (too old, or in the future), then a warning messages is added/appended to the delivery describing the problem (i.e. this is not a fatal error).
The script always exits with an exit code 0.

Sequence Diagram



Script: get_tasks_from_targetprocess.py

Purpose

This script accesses the TargetProcess API and retrieves a list of task numbers that is associated with the given product and version. The URL to the TargetProcess API and credentials is retrieved from the config manager. The task numbers can be found as a part of the title of all user stories associated with a release, where the name of the release contains the name of the product and the version given.

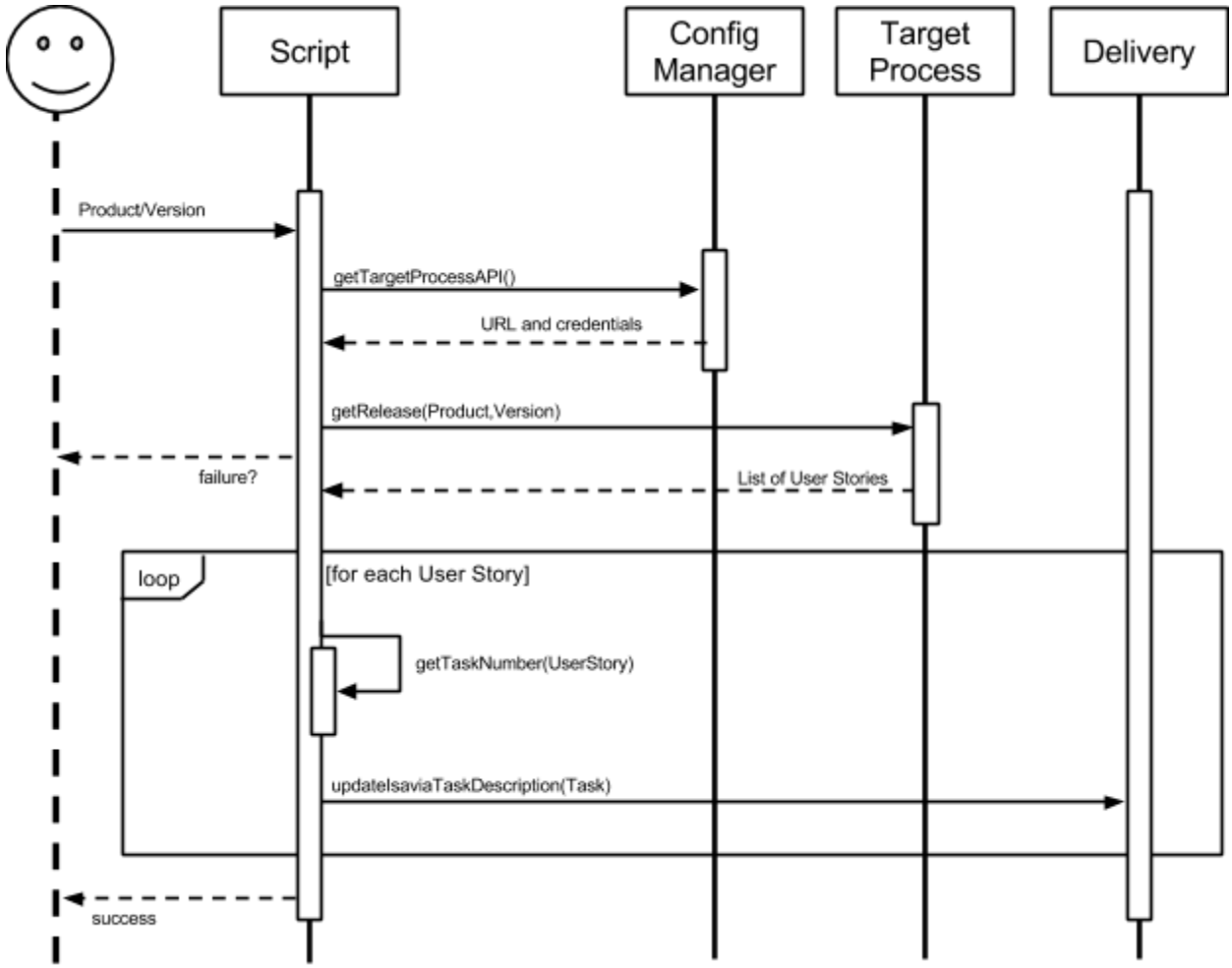
Input

This script takes an instance of a delivery as its input.

Output

If there is some error, e.g. can't connect to the TargetProcess API, then this script exits with exit code 1 and adds a relevant error message to the error_message field in the delivery. If retrieval of the task numbers was successful, then the script fills out the *isavia_task_description* field in the delivery with the tasks. In addition, it prints out the list of task numbers in a comma-separated string to the standard output and exits with exit code 0.

Sequence Diagram



Script: check_task_numbers.py

Purpose

After we have received which tasks are part of the delivery we can run this script, which will go through all the new and modified source files since the last delivery of the product currently being delivered. For each of these files, the script will run a diff against the version of the file as it was from the last delivery and look just at the new/modified lines.

If the file contains no task header, then a warning message is added to the delivery.

For each of the new task headers entries, the given task number is examined. If the task number is not in the list of task numbers previously retrieved for the delivery, then a warning message is added to the delivery.

After going through all the task header entries in all the new and modified files and there is still a task number in the list of task number that never occurred, then a warning message is added to the delivery.

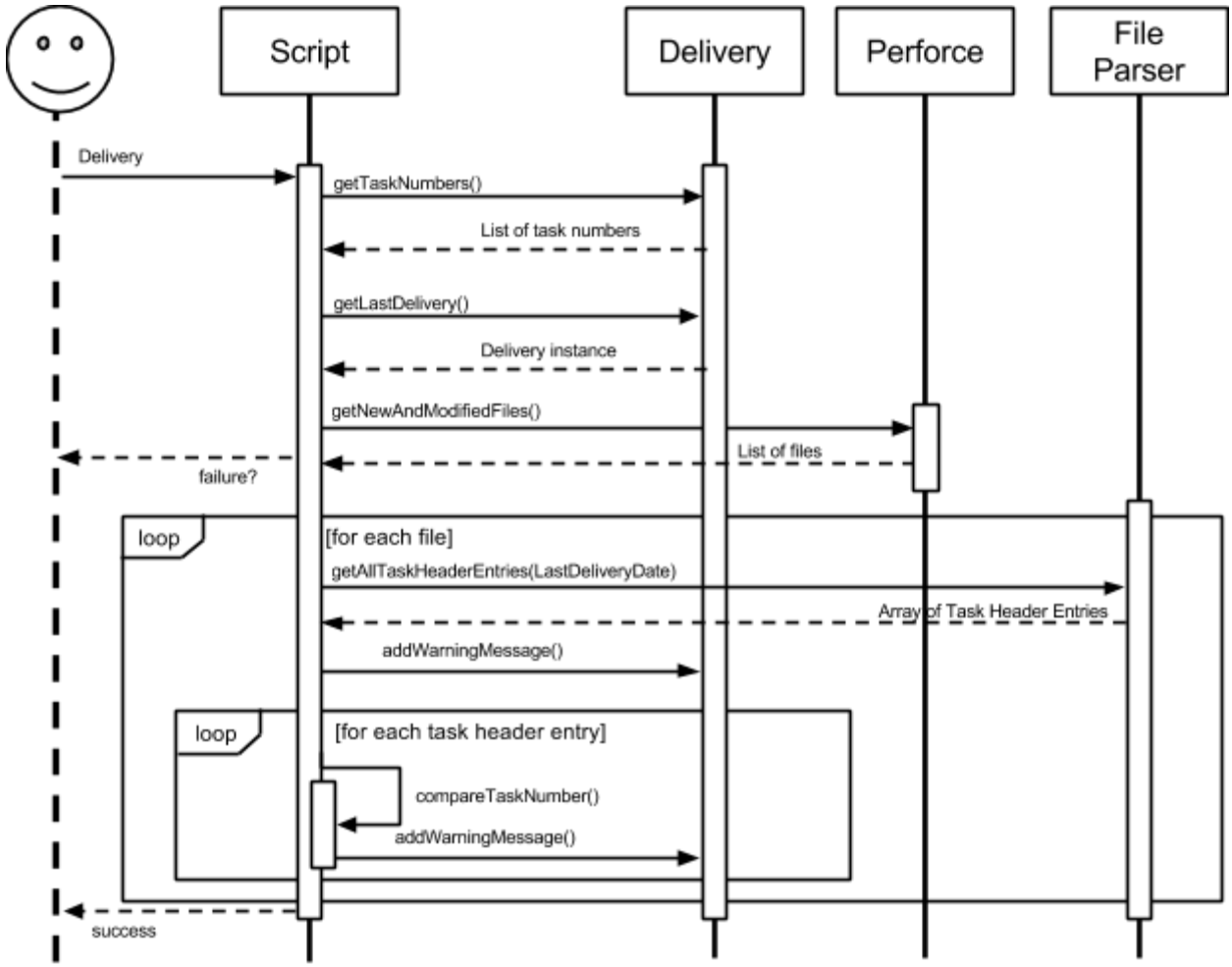
Input

This script takes an instance of a delivery as its input. A list of task numbers that are included in this delivery must have been retrieved before this script is run. This list is retrieved directly from the delivery instance. If there are no task numbers in the delivery, then a warning message is added to the delivery.

Output

This script will always exit with exit status 0. As with every script, if there are any warnings, then these will be written to the standard error file descriptor.

Sequence Diagram



Script: get_build_from_parabuild.py

Purpose

This script retrieves the build that the Parabuild Continuous Integration server has generated for the product and version that is being delivered. The URL to the server is retrieved from the config manager and an option of whether the source tarball should be retrieved as well or not is also retrieved from the config manager (this is a product-specific configuration).
The script then creates a SOAP client and uses the Parabuild web services API to find the relevant build result (one or more tarball) by matching the title of the build with the product name and version in the delivery. The tarballs are kept locally in the delivery staging directory until later when the delivery is finalized.
The script then updates the delivery with the name of the tarballs retrieved from Parabuild in the Delivery Description field.

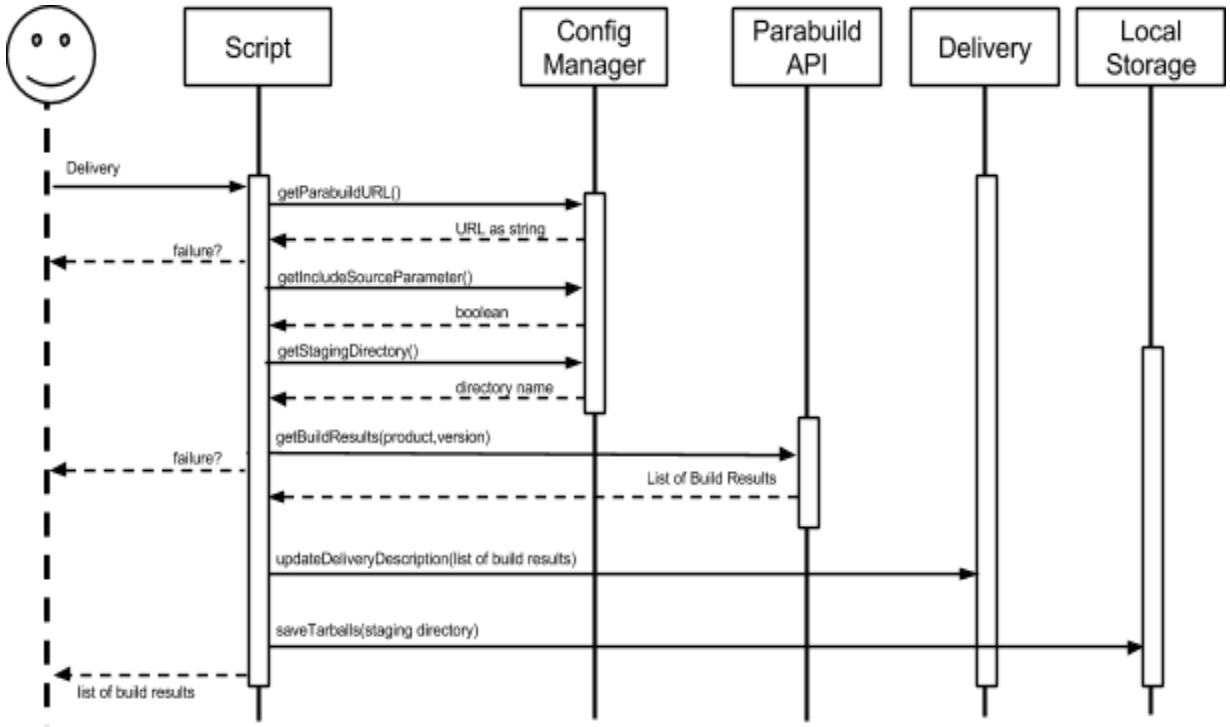
Input

This script takes an instance of a delivery as its input.

Output

The name of all the tarballs retrieved from Parabuild are joined together in a comma-separated string and printed to standard output. If the URL for the API is unreachable, or the build result is not found for the given product and version, then the script exits with error code 1 and puts the error message in the delivery.

Sequence Diagram



Script: get_documentation_from_sharepoint.py

Purpose

This script retrieves the documentation that the SharePoint server is storing for the product and version that is being delivered. The URL to the server is retrieved from the config manager and an option of whether the documentation file(s) should be converted to PDF as well or not is also retrieved from the config manager (this is a product-specific configuration). The script then creates a SOAP client and uses the SharePoint web services API to find the relevant documentation file(s) by matching the product name and version in the delivery. The documentation files are kept locally in the delivery staging directory until later when the delivery is finalized.
The script then updates the delivery with the name of the documents retrieved from SharePoint in the documentation_files field.

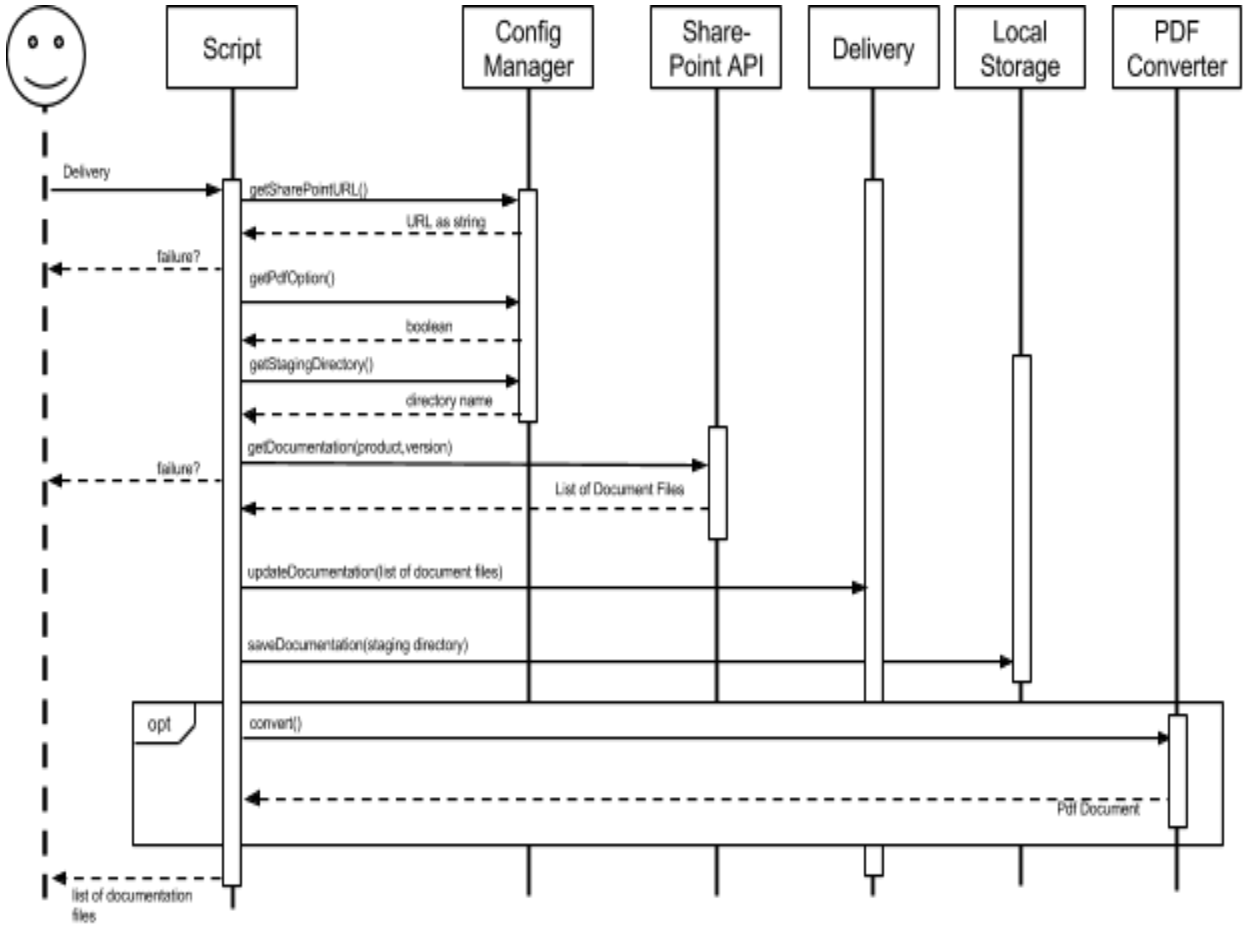
Input

This script takes an instance of a delivery as its input.

Output

The name of all the documentation files retrieved from SharePoint are joined together in a comma-separated string and printed to standard output. If the URL for the API is unreachable, or the documentation files are not found for the given product and version, then the script exits with error code 1 and puts the error message in the delivery.

Sequence Diagram



Script: create_new_mod_del_files.py

Purpose

Each delivery contains 4 files, one which lists which files are new since the last delivery, one which lists files are modified, one which lists files are removed and one file that contains the difference in the source files since the last delivery. The lists of new, modified and removed files are put in the delivery for each of their respective fields, new_source_files, modified_source_files and removed_source_files.

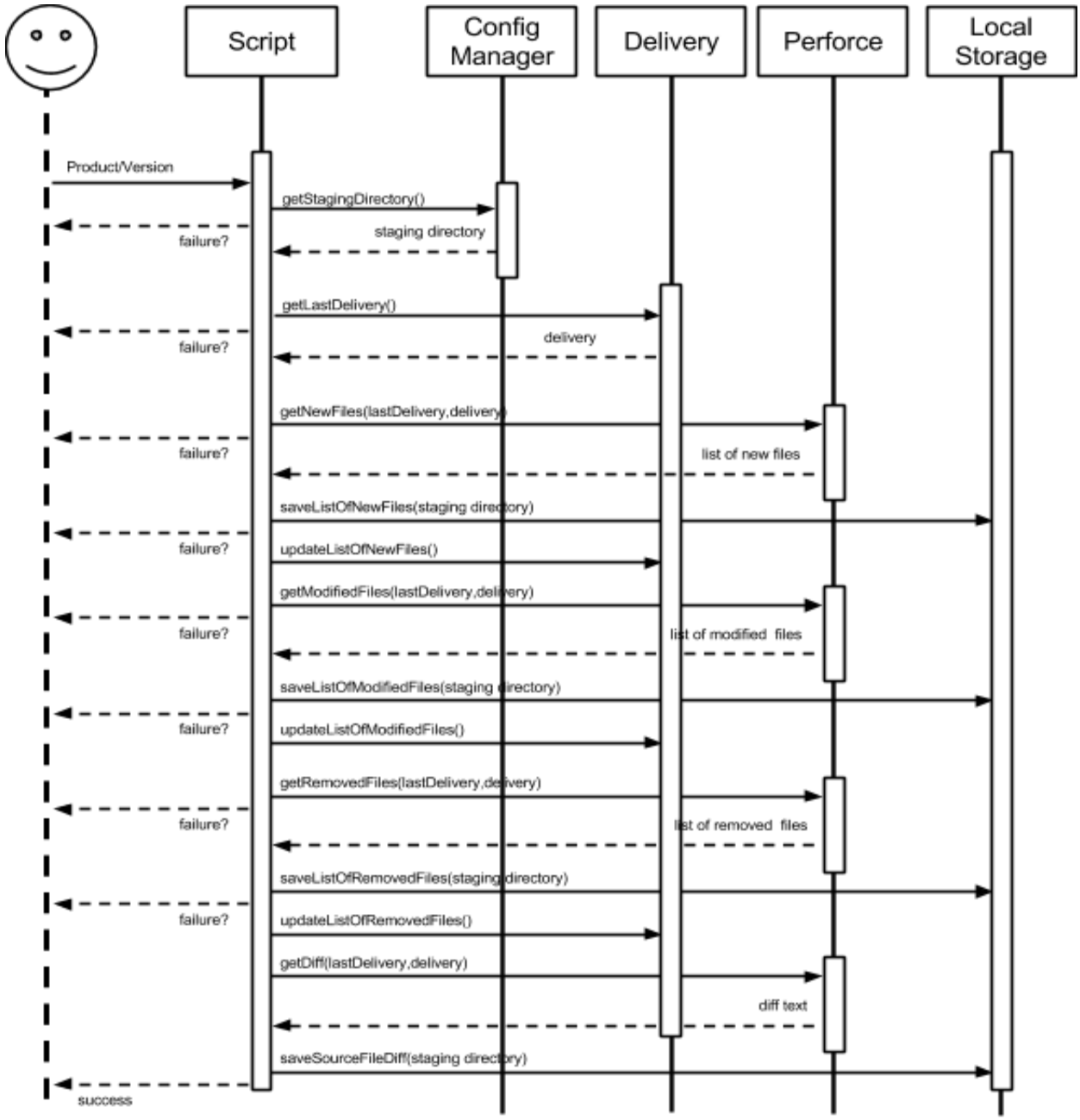
Input

This script takes an instance of a delivery as its input.

Output

This script does not output anything and exits with an exit status of 0, unless there's an error retrieving the list of new, modified and removed files, in which case an error message is put in the delivery and the script exits with exit status 1.

Sequence Diagram



Script: create_iso_and_md5.py

Purpose

After all the items of the delivery have been put together in a staging directory, this script creates an ISO file from the contents of that directory and stores the ISO file in a delivery directory. A checksum file is then created for the ISO file, which is also stored in the delivery directory.

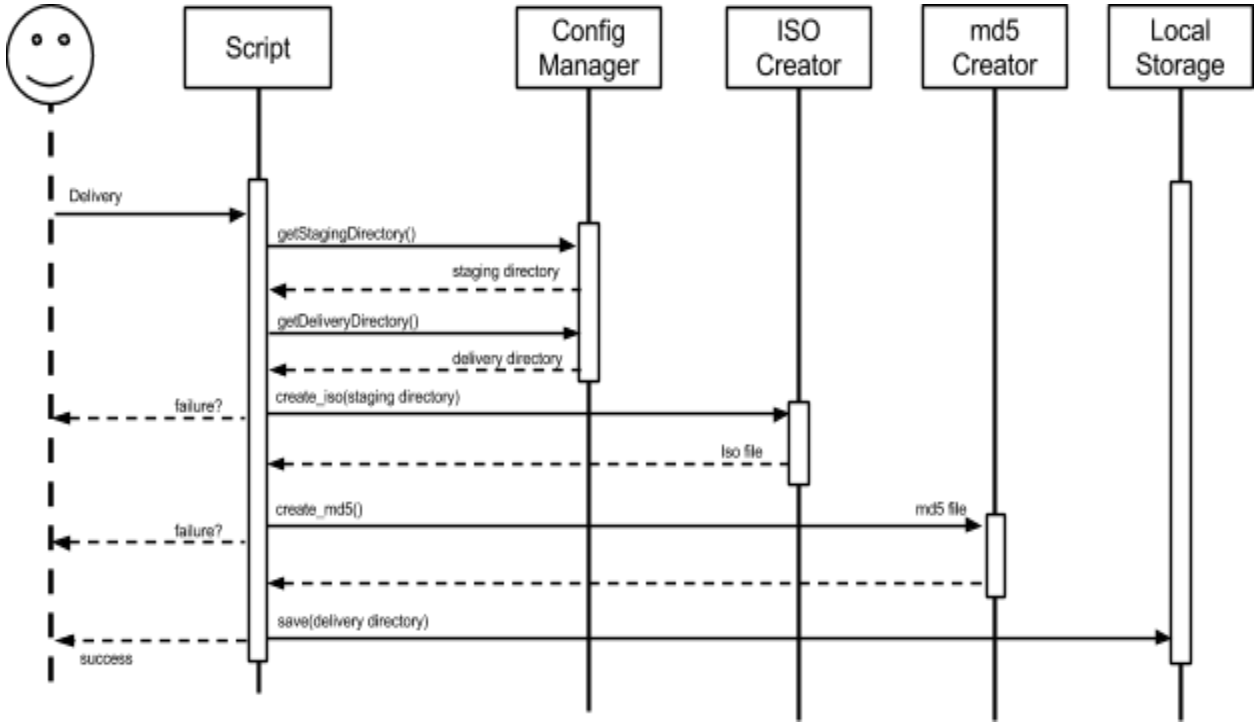
Input

This script takes an instance of a delivery as its input.

Output

Nothing is printed to standard output, and the script exits with exit code 0 unless there was a problem with creating the ISO file or checksum file (e.g. out of disk space), in which case an error message is put in the delivery and the script exits with exit code 1.

Sequence Diagram



Script: export_delivery_report_to_pdf.py

Purpose

After all the items of the delivery have been put together, this script creates a PDF file from the view of the delivery in Django and stores it in the delivery directory. The PDF is created with a Python module called pdfKit.

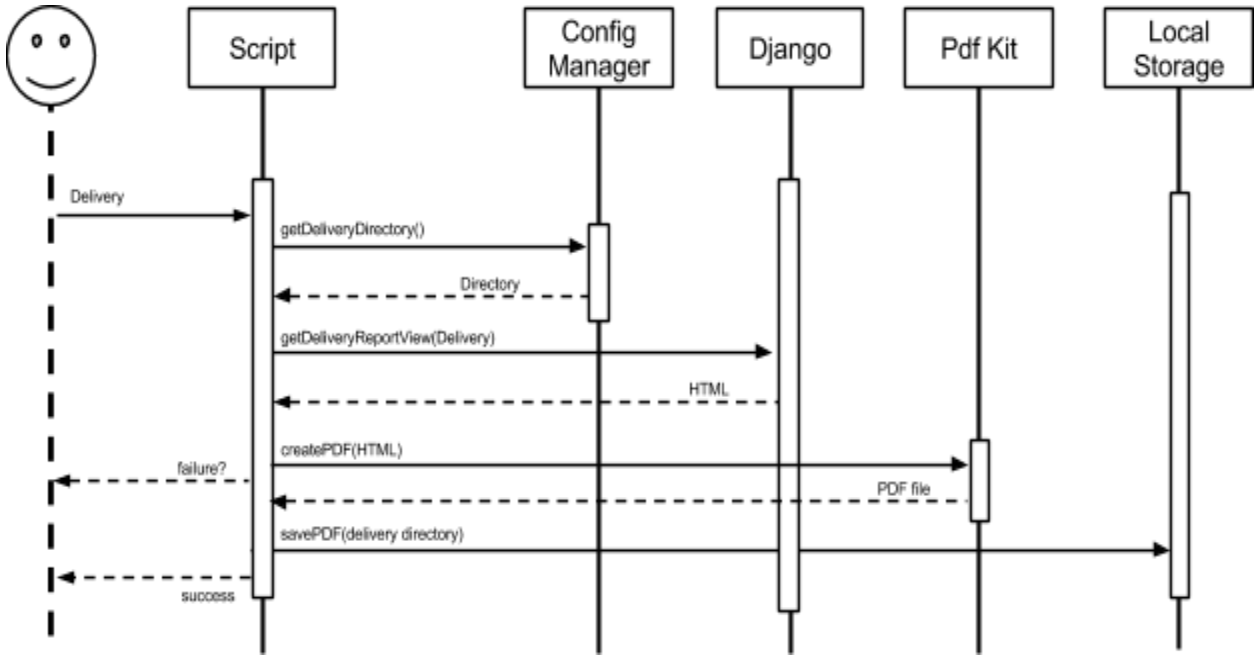
Input

This script takes an instance of a delivery as its input.

Output

Nothing is printed to standard output, and the script exits with exit code 0 unless there was a problem with creating the PDF file (e.g. out of disk space), in which case an error message is put in the delivery and the script exits with exit code 1.

Sequence Diagram



Script: copy_delivery_files_to_storage_server.py

Purpose

After the delivery files are ready, this script takes them and copies to a server for storage/backup. The hostname of the server and credentials are retrieved from the config manager. The files are then uploaded using scp.

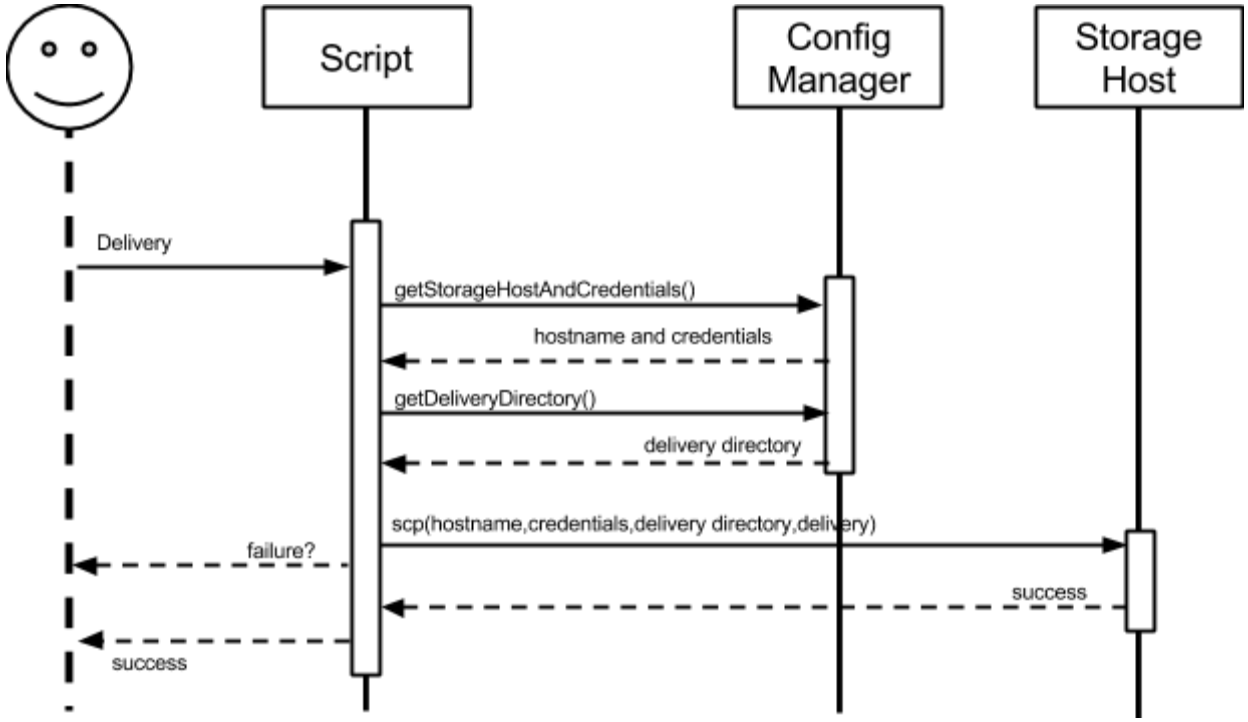
Input

This script takes an instance of a delivery as its input.

Output

Nothing is printed to standard output, and the script exits with exit code 0 unless there was a problem with uploading the delivery files (e.g. host not found), in which case an error message is put in the delivery and the script exits with exit code 1.

Sequence Diagram



Script: task_header_parser.py

Purpose

Input

Output

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

Tern Systems - Working Arrangements (Verkskipulag)

T-404-LOKA

Students:
Árni Þorvaldsson (arnitho12@ru.is)
Freyr Bergsteinsson (freyrb12@ru.is)
Gunnar Þór Helgason (gunnarh07@ru.is)
Sigurbjörn Kristjánsson (sigurbjorn12@ru.is)

Instructor:
Daníel Máni Jónsson (daniel@valitor.is)

Examiner:
Hannes Pétursson (hap@ru.is)

Table of Contents

Document History

Date	Author	Description
2015-01-27	ÁÞ, FB, GÞH, SK	Initial version, based in parts on the Project Report, delivered on 20/1/2015.
2015-03-10	ÁÞ, FB	Minor revision, finished all sprints and updated which reports are turned in.

Introduction

Work on the project will be done in compliance with Agile methodology, specifically using Scrum. The team that will work on the project has been given the name *Autobots*.

Roles and responsibilities

Scrum Master

The Scrum Master of the team is Freyr Bergsteinsson <freyrb12@ru.is>. It was decided he would take this role because of previous Agile experience, even though he has more experience with Kanban than with Scrum. In addition to being a student, Freyr is also an employee at Tern Systems, where both Kanban and Scrum is being used.

Product Owner

The Product Owner is Heiðar Harðarson <heidar@tern.is>, which is also the team's correspondent at Tern Systems. Mr. Harðarson has been employed by Tern Systems since 2000 as a programmer and project manager. He leads the team at Tern Systems that is responsible for development and maintenance of systems owned by Isavia. This gives him a very clear vision in regards to what the end-product of this project should feature.

Team

The entire Autobots team consists of the following members:

- Árni Þorvaldsson <arnitho12@ru.is>
- Freyr Bergsteinsson <freyrb12@ru.is>
- Gunnar Þór Helgason <gunnarh07@ru.is>
- Sigurbjörn Kristjánsson <sigurbjorn12@ru.is>

Each member has the role of a developer, which is an interdisciplinary role, meaning everyone takes on tasks that pertain to coding, documenting, testing and more.

Sprint arrangements

The duration of the first Scrum sprint, Sprint Zero, was 7 days, from January 19th to January 26th beginning and ending on a Monday.

The succeeding sprints will have a duration of 14 days, also beginning on and ending on Mondays, with sprint review (including a demo), retrospective analysis and planning during the meetings.

Sprint #	Sprint Name	Date Start	Date End
0	N/A	2015-01-19	2015-01-26
1	Grimlock	2015-01-26	2015-02-09
2	Bumblebee	2015-02-09	2015-02-23
3	Megatron	2015-02-23	2015-03-09
4	Ironhide	2015-03-09	2015-03-24
5	Jazz	2015-03-24	2015-04-06
6	Optimus Prime	2015-04-06	2015-04-20
7	Hot Rod	2015-04-20	2015-05-04
8	Mirage	2015-05-04	2015-05-11
9	Ultra Magnus	2015-05-11	2015-05-18

The length of sprints 8 and 9 are 7 days instead of 14 days, since all members of the team will have more time to work on this project from that point on.

Scheduled meetings

Team meetings are usually held at Tern Systems' premises at least three times a week, for about 4-6 hours at a time. Regular meeting are on-site on Mondays from 12:30-17:30 and on Tuesdays from 19:30-23:30. Other meeting times are scheduled on a per-week basis and will be either on-site or off-site depending on what's being worked on.

Daily meetings will be held at the start of the regular meetings on Mondays and Tuesdays. On Wednesdays, Thursdays and Fridays the team will have daily meetings through Google Hangout at 14:00. No daily meetings are scheduled for Saturdays or Sundays, although they will be scheduled if the need calls for it.

Starting halfway through Sprint 7, the team will meet every day for 8 hours at a time until the end of Sprint 9. Capacity will then be adjusted and those sprints planned accordingly with regards to the calculated team velocity.

Reports

In addition to the software developed for Tern Systems, the end product for the project itself will include the following reports:

Report	**Hand-in date**
Project Report (Verkefnislýsing)	2015-01-20 (done)
Procedural Description (Verklagslýsing)	2015-01-27 (done)
Working Arrangement (Verkskipulag) - This report	2015-01-27 (done)
Project Schedule (Verkáætlun)	2015-02-10 (done)
Risk Analysis (Áhættugreining)	2015-02-10 (done)
Reports for Status Meeting 2. Includes updated versions of all previous reports, as well as ● Design Document (Hönnunarskýrsla) ● Operational Manual (Rekstrarhandbók) ● User Guide (Notendaleiðbeiningar)	2015-03-22 (done)
Reports for Status Meeting 3. Includes updated versions of all previous reports.	2015-05-04 (done)
Final report (Lokaskýrsla). Includes updated versions of all previous reports compiled into a single report. Also includes ● Work Journal (Vinnudagbók)	2015-05-14

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

Tern Systems - Development Report (Framvinduskýrsla)

T-404-LOKA-2015

Students:
Árni Þorvaldsson (arnitho12@ru.is)
Freyr Bergsteinsson (freyrb12@ru.is)
Gunnar Þór Helgason (gunnarh07@ru.is)
Sigurbjörn Kristjánsson (sigurbjorn12@ru.is)

Instructor:
Daníel Máni Jónsson (daniel@valitor.is)

Examiner:
Hannes Pétursson (hap@ru.is)

Table of contents

Sprint planning
Sprint burndown
Team velocity
Sprint review

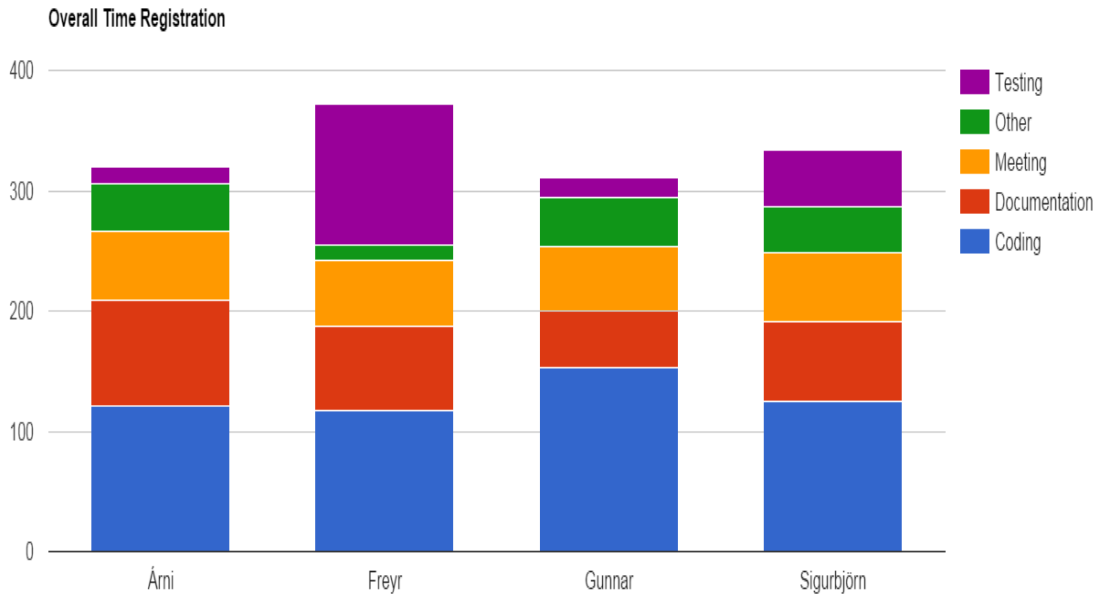
Document History

Date	Author	Description
2015-02-15	ÁÞ, FB, GÞH, SK	Initial version.
2015-03-10	ÁÞ	Inserted Sprint 3 and sprint retrospective and time registration for Sprint 2
2015-03-29	ÁÞ	Inserted sprint 4 (added burn-up chart and team's capacity). Added backlog for sprint 5. Updated overall time registration and updated overview of on-going and completed sprints.
2015-04-08	ÁÞ	Updated overall time registration and burn-up chart. Updated sprint 5. Added backlog for sprint 6.
2015-04-27	ÁÞ	Updated overall time registration and burn-up chart. Updated sprint 6. Added backlog for sprint 7
2015-05-04	ÁÞ	Updated overall time registration and burn-up chart. Updated sprint 7. Added script status chart and cumulative flow chart for user stories.
2015-05-13	ÁÞ	Updated charts, updated sprint 8 and added sprint 9.

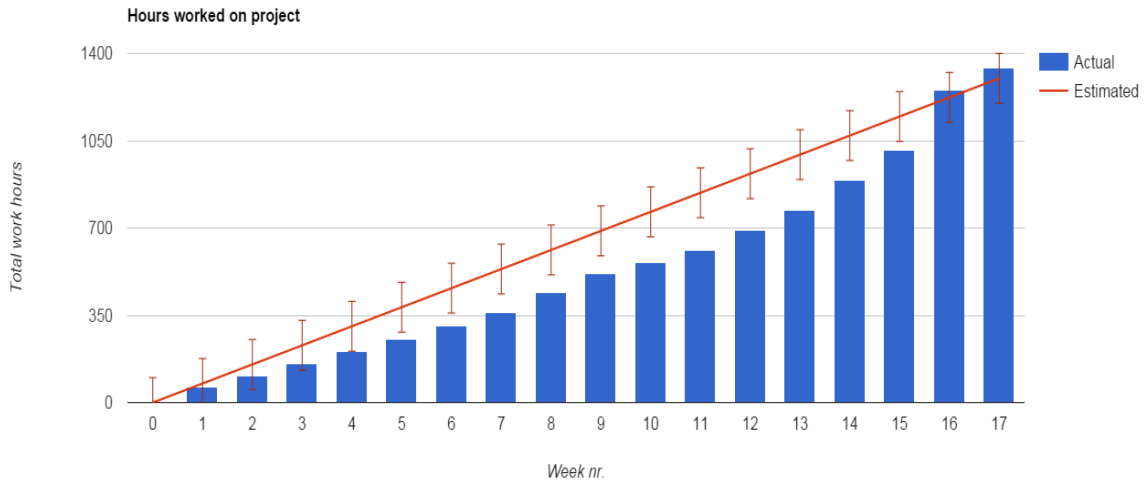
Overall Time Registration

(last update: May 13th)

	Coding	Documentation	Meeting	Other	Testing	Total
Árni	121.5	88	56.5	40	13.5	319.5
Freyr	117.5	70	55	12	117	371.5
Gunnar	153.5	46	54.5	40	17	311
Sigurbjörn	124.5	66.5	57	38	48	334
Total	517	270.5	223	130	195.5	1336

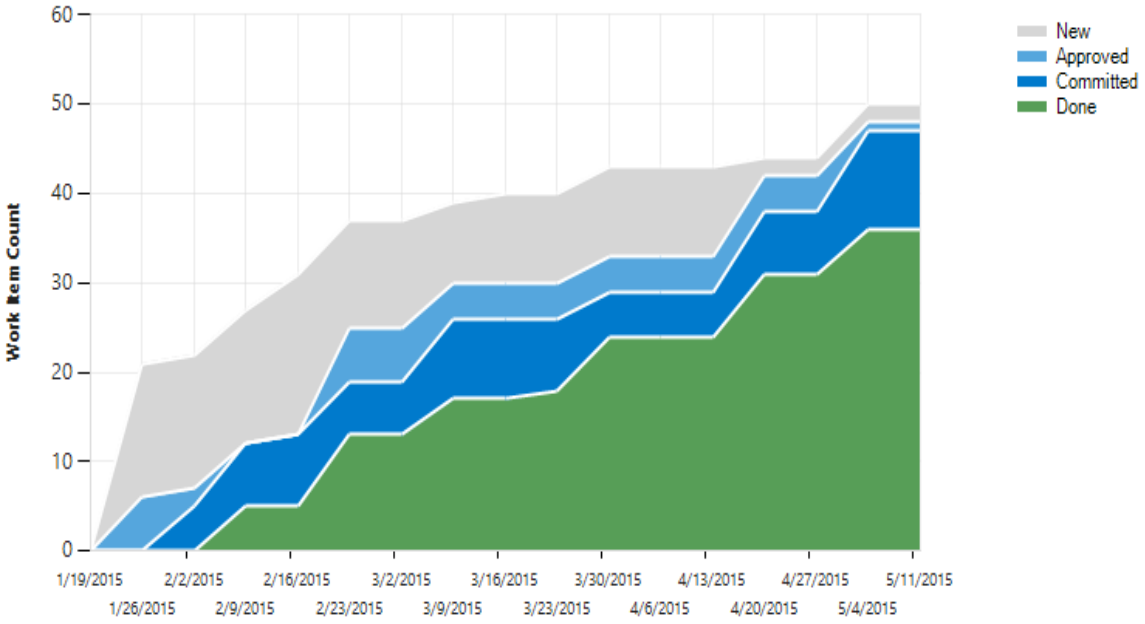


Current burn-up chart (updated May 13th)

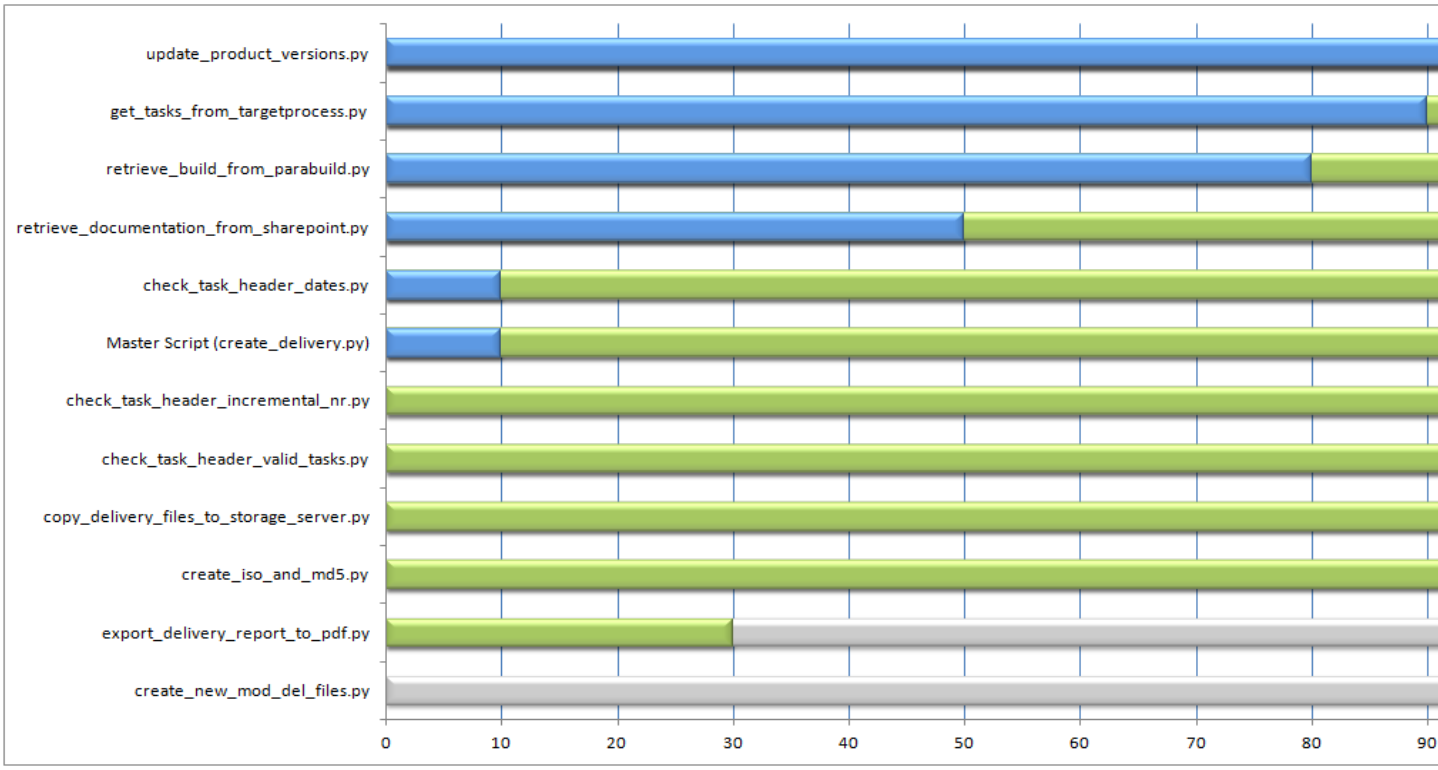


Project Progress

Cumulative flow of user stories in backlog (updated May 11th)



Script status (updated May 7th)



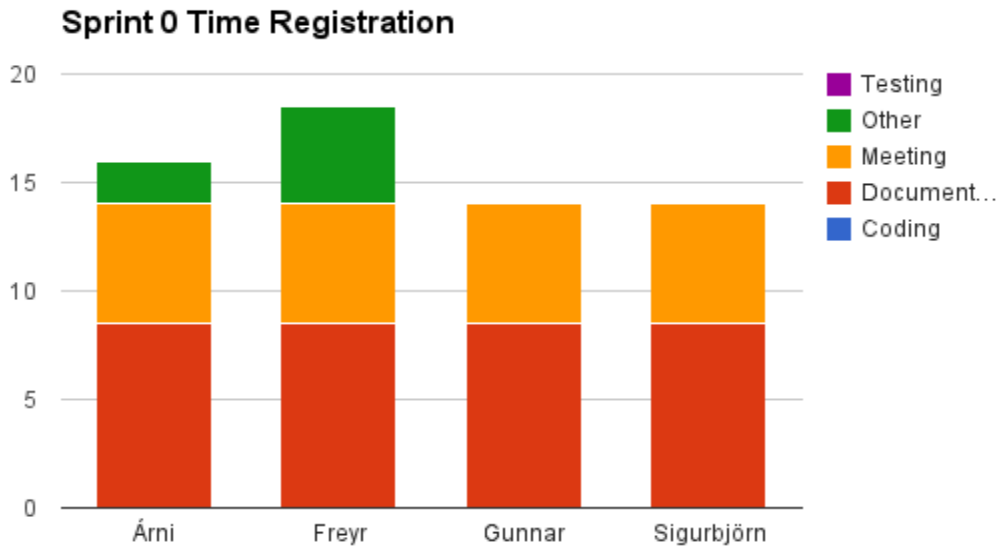
Overview of completed and on-going sprints

Sprint #	Sprint Name	Date Start	Date End
0	N/A	2015-01-19	2015-01-26
1	Grimlock	2015-01-26	2015-02-09
2	Bumblebee	2015-02-09	2015-02-23
3	Megatron	2015-02-23	2015-03-09
4	Ironhide	2015-03-09	2015-03-23
5	Jazz	2015-03-24	2015-04-05
6	Optimus Prime	2015-04-08	2015-04-19
	/* Time off */	2015-04-21	2015-04-26
7	Hot Rod	2015-04-27	2015-05-03
8	Mirage	2015-05-04	2015-05-10
9	Ultra Magnus	2015-05-11	2015-05-17

Sprint 0

Sprint 0 was a week long and was mostly seen as an introduction to the project and for choosing and setting up software and tools. The team agreed on using the Scrum software development framework for managing product development. First draft of project report was completed.

Time registration



Sprint 1 (Grimlock)

Before sprint planning the team wrote user stories and estimated their story points by planning poker. The user stories were then assigned to the product backlog.

Sprint planning

At sprint planning the team chose user stories for the sprint and broke them into tasks. Each tasks' working hours was then roughly estimated.

Sprint backlog for Sprint 1	Est.
As an instructor and examiner I want to see the team's working arrangements (Verkskipulag) so I can assess the progress of the project	1 sp
• Create report for working arrangements	10 hrs

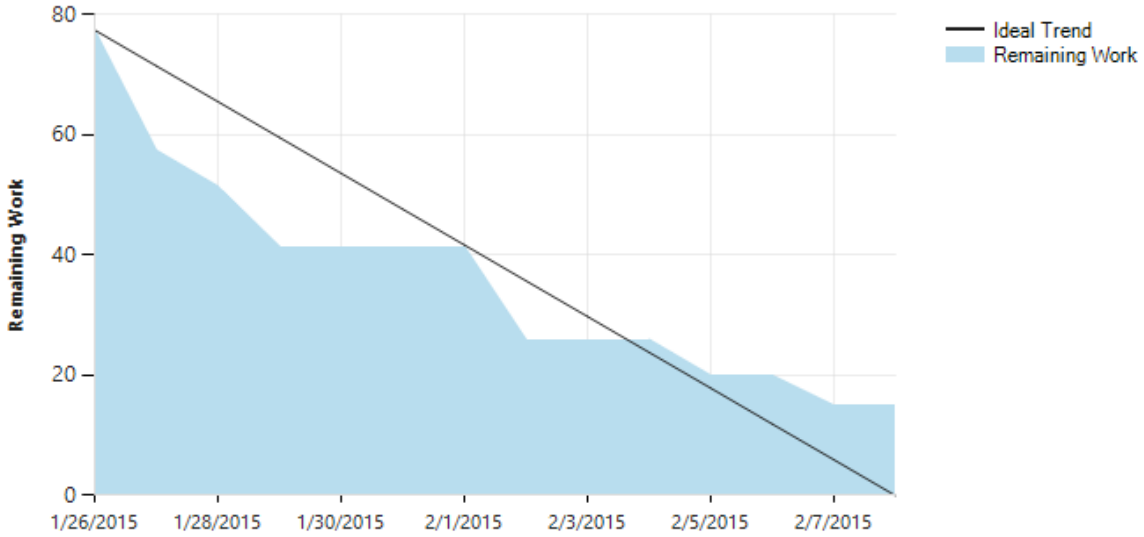
As an instructor and examiner I want to see the team's Project Schedule (Verkáætlun) so I can assess the progress of the project	1 sp
• Create a report for Project Schedule	20 hrs
As an instructor and examiner I want to see a presentation of the product so I can assess the progress of the project	1 sp
• Create a presentation	10 hrs
As as user I want a web front end to the product so that I can interact with it	2 sp
• Set up Django on the product server	2 hrs
• Create HTML for the web page	2 hrs
As a developer I want a server for the product so that I can run a centralized server for web front end and running of scripts	2 sp
• Request a virtual machine from Tern's system administrator	2 hrs
• Install Fedora 20/21 on server	4 hrs
• Install MySQL on server	2 hrs
• Install Python on server	0.5 hrs
• Install Beautiful Soup 4 on server	0.5 hrs
• Install Git client on the server	0.5 hrs
• Install LaTeX on server	4 hrs
As a scrum master I want to see scrum tools in order so that we can report work progress (Framvinduskýrsla)	1 sp
• Investigate how planning poker should take place	0.5 hrs
• Move all backlog items from Trello to VSO	0.5 hrs
• Display a burndown chart for the current sprint	0.5 hrs

The following user story was removed from the sprint due to updated requirements:

As a user I want to configure the tool to run appropriate tasks	8 sp
• Create test for config manager	4 hrs

• Create a table in a database that contains all config values	4 hrs

Sprint burndown for Sprint 1



The remaining work does not end at 0 due to one user story being removed from the sprint after the sprint was over (Visual Studio Online does not reflect this retroactively).

Sprint retrospective

The team finished 11 story points in the first sprint, so the initial velocity of the team is 11 story points. The sprint shows user stories for a total of 8 story points, but an estimated 3 further story points were done on a big (8 story point) user story, which was moved to sprint 2.

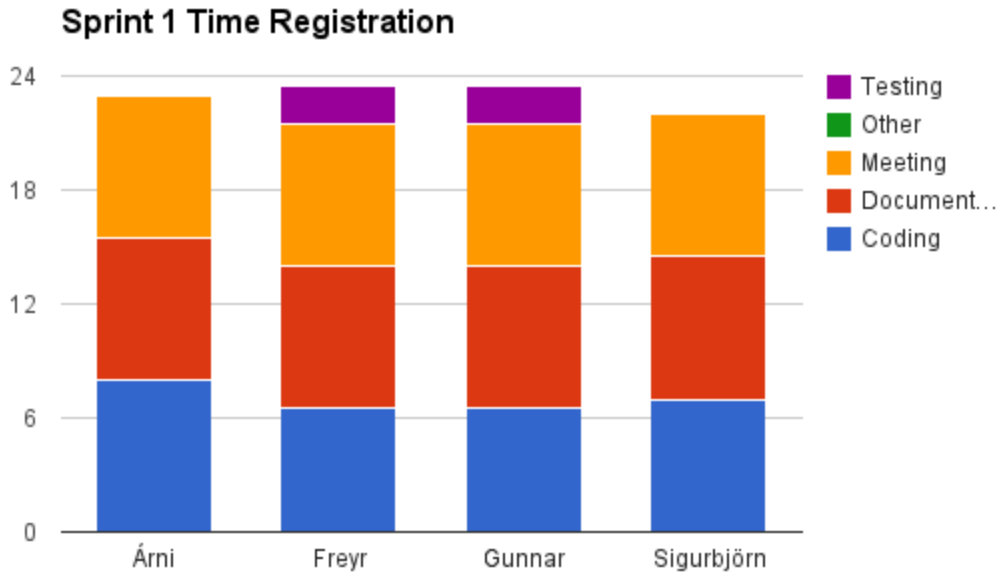
What went well during the sprint?

Work on reports and presentation went well.

What could be improved in the next sprint?

The team agreed that it might benefit from drawing diagrams before starting working on tasks that require coding, to enhance understanding visually of what is required of those tasks. Might be useful to examiner and instructor as well.

Time registration



Sprint 2 (Bumblebee)

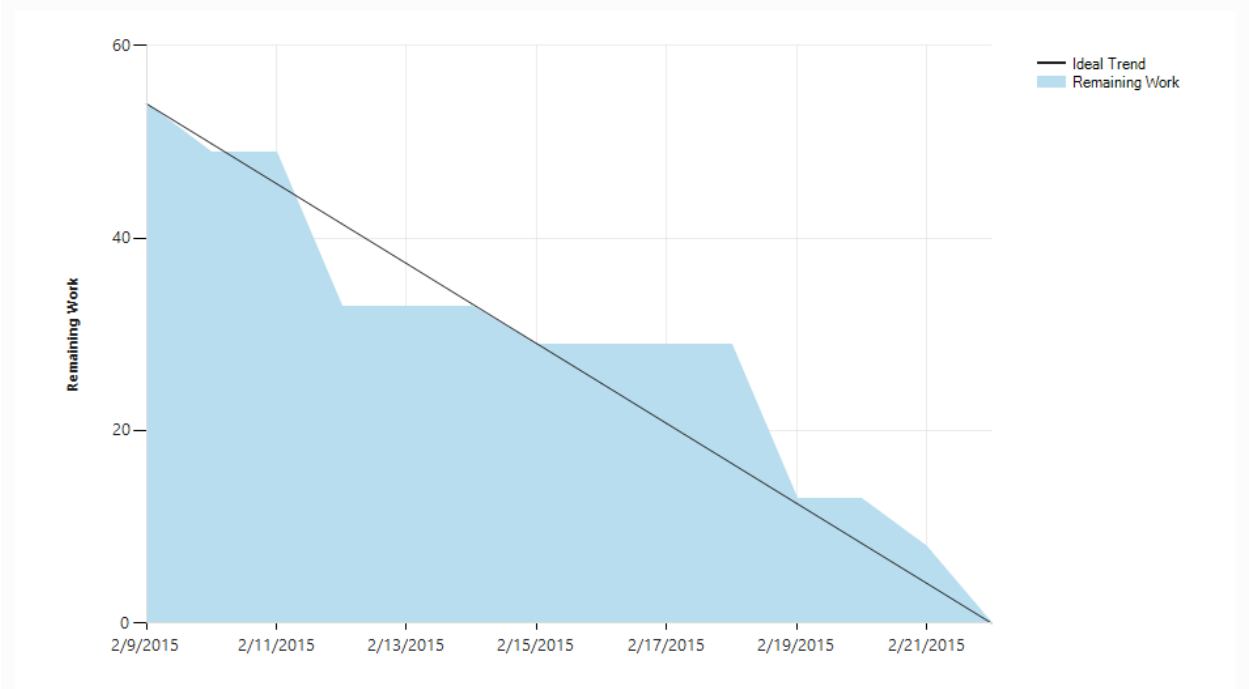
Sprint planning

User stories were selected for the sprint with the velocity of the previous sprint in mind. The user stories were then broken into the following tasks. A rough estimation of working hours was made as well.

Sprint Backlog for Sprint 2	Est.
As an instructor and examiner I want to see the team's Developments Report (Framvinduyfirlit) so I can assess the progress of the project	2 sp
● Create a Development Report (Framvinduskýrsla)	12 hrs
● Show how much time has been spent in total, per type and per member	2 hrs
As an instructor and examiner I want to see the team's Risk Analysis (Áhættugreining) so I can assess the progress of the project	2 sp
● Create a Risk Analysis Report (Áhættugreining)	12 hrs
As an instructor and examiner I want to see the team's updated Project Schedule (Verkáætlun) so I can assess the progress of the project	1 sp
● Update the Project Schedule (Verkáætlun)	10 hrs
As a developer I want a better overview of the internal working of the product so that I better understand it	3 sp
● Create a schema of the database for the configuration manager	2 hrs
● Create a schema of the delivery report database	3 hrs
● Create a sequence diagram of the master script	6 hrs
As a user I want to make a master script that runs the whole project so that it is simpler to create delivery	5 sp
● Create a Python script that takes product and version as an input	8 hrs
● Let the script return a warning if something is wrong but doesn't stop the delivery process	2 hrs
● Let the script return a success if delivery procedure went well	2 hrs
● Let the script return an error with an error message if something went wrong during the delivery procedure	2 hrs

As an instructor and examiner I want to see a presentation of the product for the first status meeting (Stöðufundur 1) so I can assess the progress of the project	1 sp
• Update the presentation used the last time to include risk analysis and development report	4 hrs
• Rehearse for the presentation	2 hrs

Sprint burndown for Sprint 2



Sprint 2 retrospective

What went well

- Better workflow than previous sprint
- Status meeting very informative
- Good spirits within team

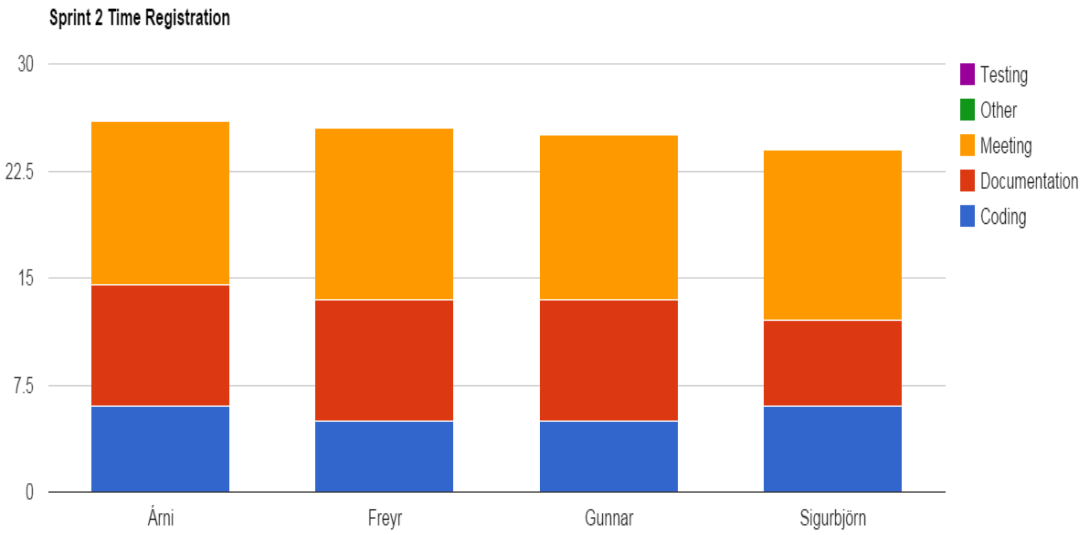
What could be done better

- Explain product better to examiner
- Team members should be more vigilant in asking for help
- Document what to do in more detail before coding
- Work journal entries disappeared

Actions

- Update documentation with points from status meeting
- Update work journal summary to better handle entries, preferably so that entries can be assigned to sprints by date alone (this way the position of the entry does not matter)

Time registration



Sprint 3 (Megatron)

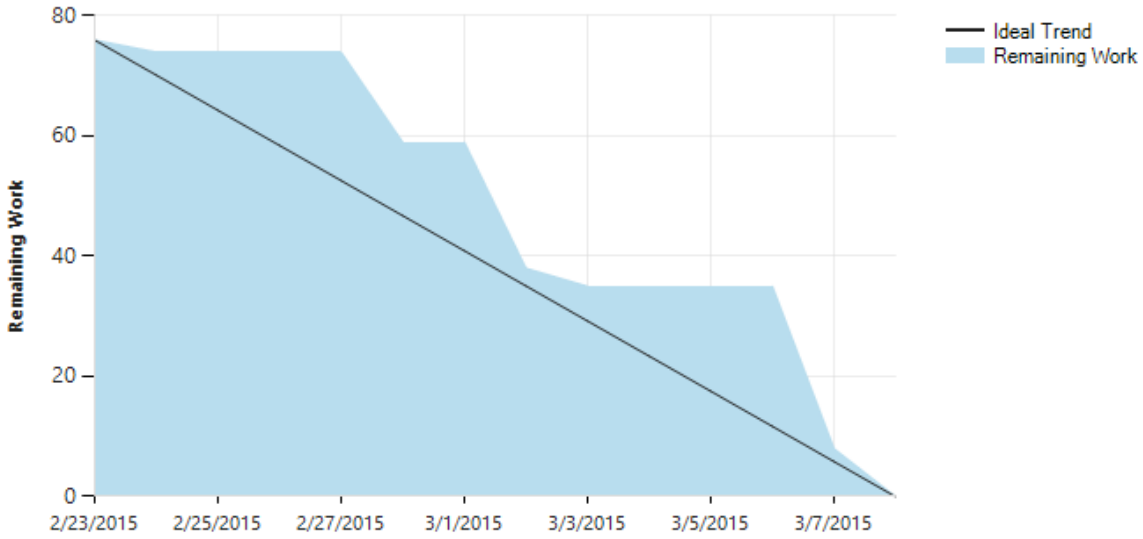
Sprint planning

User stories were selected for the sprint with the velocity of the previous sprint in mind. The user stories were then broken into the following tasks. A rough estimation of working hours was made as well.

Sprint Backlog for Sprint 3	Est.
As a developer I want to easily add unit tests to the product so that the product is more reliable	2 sp
● Investigate how unit tests are made with Django	3 hrs
As a developer I want the config manager in a separate database than the delivery report as to keep the tool separate from the data	2 sp
● Investigate how to make Django use different databases	3 hrs
● Split delivery report model from config manager model	2 hrs
● Split config manager into two apps, one for the config values and one for the product catalog	1 hr
As a developer I want the Config Manager to be part of the Django framework so that it is easier to maintain	4 sp
● Implement unit test for add function in the config manager	3 hrs
● Implement unit test for the get function in the config manager	3 hrs
● Implement unit test for the update function in the config manager	3 hrs
● Implement unit test for the delete function in the config manager	3 hrs
● Implement add function in the config manager that stores a new config value	2 hrs
● Implement get function in the config manager that retrieves a value for some key	2 hrs
● Implement update function that updates the value for some key in the config manager	3 hrs

• Implement a delete function in the config manager that removes a key/value pair from the database	2 hrs
• Implement unit test for the add function for products in the config manager	1 hr
• Implement unit test for the update function for products in the config manager	1 hr
• Implement unit test for the delete function for products in the config manager	1 hr
• Implement the add function for products in the config manager	1 hr
• Implement the update function for products in the config manager	1 hr
• Implement the delete function for products in the config manager	1 hr
• Implement unit test for updating with addIfNotExist flag set to true	1 hr
As a user I want delivery reports to be stored in a database so I can use it to generate a report later	2 sp
• Implement createNewDelivery function in the config manager that adds a new entry in the delivery report database	12 hrs
As an instructor and examiner I want to see the project documents updated with regards to points made in Status Meeting 1	1 sp
• Update risk analysis and split the "work hindered" into more specific risks	2 hrs
• Update estimated work hours to include additional requirements (300-350 hrs / person)	2 hrs
As an instructor and examiner I want to see a draft of a design document so that I can better understand how the product is supposed to work	3 sp
• Update sequence diagram for the master script	2 hrs
• Create a diagram for script that retrieves available products and their versions	2 hrs
• Create a diagram for script that checks if task header dates are ok	2 hrs
• Create a database schema for the product table, config table and delivery table	3 hrs
• Create a diagram for script that checks if task header numbers (incremental) are ok	2 hrs

Sprint burndown for Sprint 3



Sprint retrospective

What went well

- Better overview of the project than before
- Unit tests went well
- Diagrams help to visualize and deepen understanding

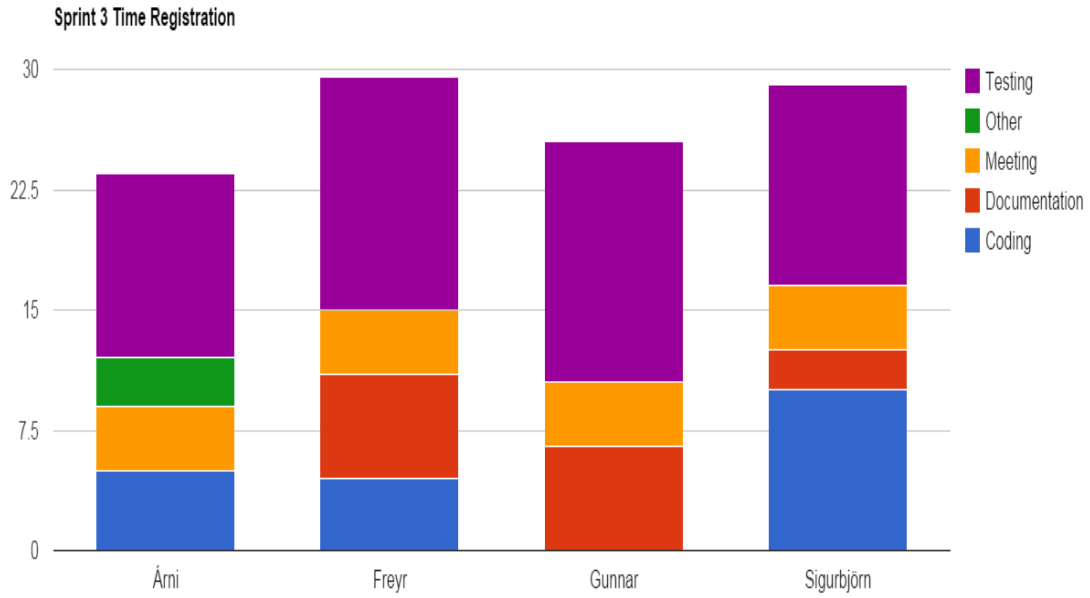
What could be done better

- Encourage dialogue between group members - ask questions

Actions

- Setup virtual environment?
- Setup Django locally on all computers

Time registration



Sprint 4 (Ironhide)

Sprint planning

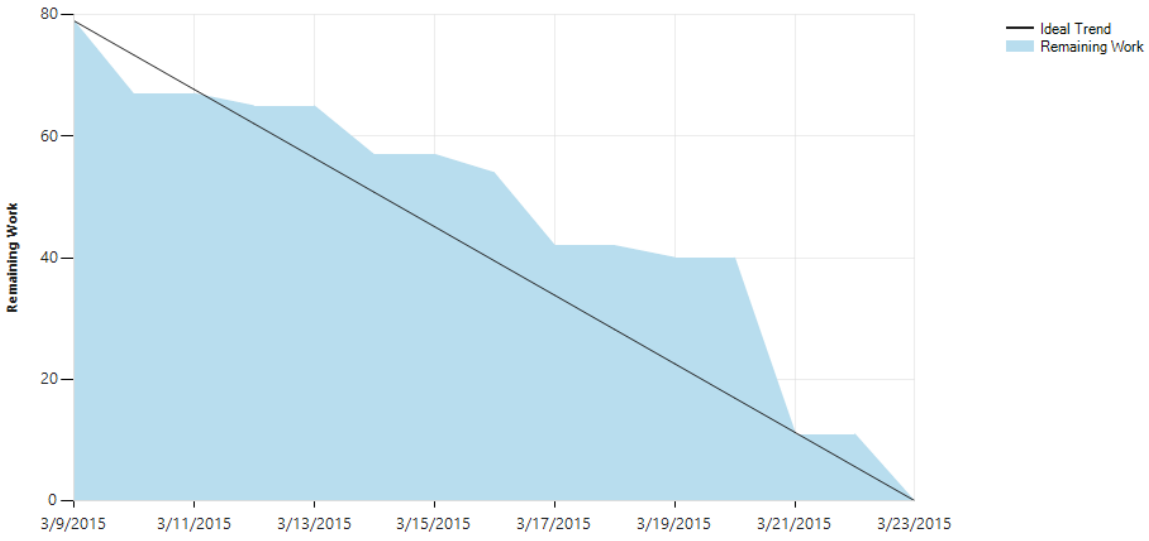
User stories were selected for the sprint with the velocity of the previous sprint in mind. The user stories were then broken into the following tasks. A rough estimation of working hours was made as well.

Sprint Backlog for Sprint 4	Est.
As a user I want to be able to look at and edit data in an admin web interface to better control the data in a visual manner	1 sp
● Add the current models to the admin interface in the Django project	1 hr
As a user I want delivery reports to be stored in a database so I can use it to generate a report later	2 sp
● Implement unit test for the Delivery model	4 hrs
● Have the master script generate a new delivery instance	2 hrs
As an instructor and examiner I want to see a draft of a design document so that I can better understand how the product is supposed to work	3 sp
● Create a diagram for script that retrieves list of tasks from JIRA/TargetProcess	2 hrs
● Create a diagram for script that checks if all task numbers (and only those task numbers) are in the delivery	2 hrs
● Create a diagram for script that retrieves product build	2 hrs
● Create a diagram for script that retrieves documentation	2 hrs
● Create diagram for script that exports delivery report to PDF	2 hrs
● Create diagram for script that creates the ISO and MD5 files	2 hrs
● Create a diagram for the master script that shows how it handles warnings and errors	2 hrs
● Create diagram for script that copies the ISO, MD5 and PDF files to a storage server	2 hrs

• Create a diagram for script that creates files of new, modified and deleted files.	1 hrs
As a user I want to fetch a list of projects and versions from Perforce so I can use them to select what to deliver	3 sp
• Ask Tern system administrator for Perforce credentials that only DATI will be using	1 hr
• Have update_product_versions.py retrieve Perforce credentials from the config manager	2 hrs
• Have update_product_versions.py retrieve a list of paths from Perforce to get a list of products	2 hrs
• For each path, retrieve a list of product from Perforce	4 hrs
• Update product database with list of products retrieved from Perforce	3 hrs
• For each product, retrieve a list of product versions from Perforce	3 hrs
• Update product versions database with list of product versions retrieved from Perforce	3 hrs
• Have update_product_versions.py return the list of product versions as a string	2 hrs
As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	1 sp
• Update development report (framvinduskýrsla) after sprint 3	2 hrs
As an instructor and examiner I want to see a presentation of the product for the second status meeting (Stöðufundur 2) so I can assess the progress of the project	1 sp
• Add to design document examples of different task headers	2 hrs
• Update presentation	10 hrs
• Update and prepare Project Report (Verkefnislýsing) for status meeting 2	3 hrs
• Update and prepare Working Arrangement (Verkskipulag) for status meeting 2	1 hr
• Update and prepare Project Schedule (Verkáætlun) for status meeting 2	2 hrs
• Update and prepare Risk Analysis (Áhættugreining) for status meeting 2	1 hr

• Create a draft for Operational Manual (Rekstrarhandbók) for status meeting 2	1 hr
• Create a draft for User Guide (Notendaleiðbeiningar) for status meeting 2	1 hr
• Update and prepare Design Document (Hönnunarskýrsla) for status meeting 2	10 hrs
• Update and prepare Work Journal (Vinnudagbók) for status meeting 2	2 hr
As a user I want to automatically get user manuals for a specific version of my software from SharePoint so I can add it to the delivery	3 sp
• Ask Tern system administrator for SharePoint credentials that only DATI will be using	1 hrs
• Retrieve SharePoint credentials from the config manager	2 hrs
• Have script log in to SharePoint with credentials	1 hrs
• Retrieve URLs for documents for the given product from the config manager	2 hrs
• Retrieve documents from SharePoint with the URLs retrieved from the config manager	3 hrs

Sprint burndown for Sprint 4



Team's velocity

The team's velocity for sprint 4 was **11.5 story points**. Total story points were 14, as two stories, unfinished from sprint 3, were assigned to sprint 4.

Retrospective after Stöðufundur 2

What went well

- generally positive meeting
- few questions, instructors seemed to get the point of the project

What could be done better

- better preparation
- explain which scripts are completed and what is left to-do

Actions

- deal with request time-out from browser
- Use RabbitMQ?

Sprint 4 retrospective

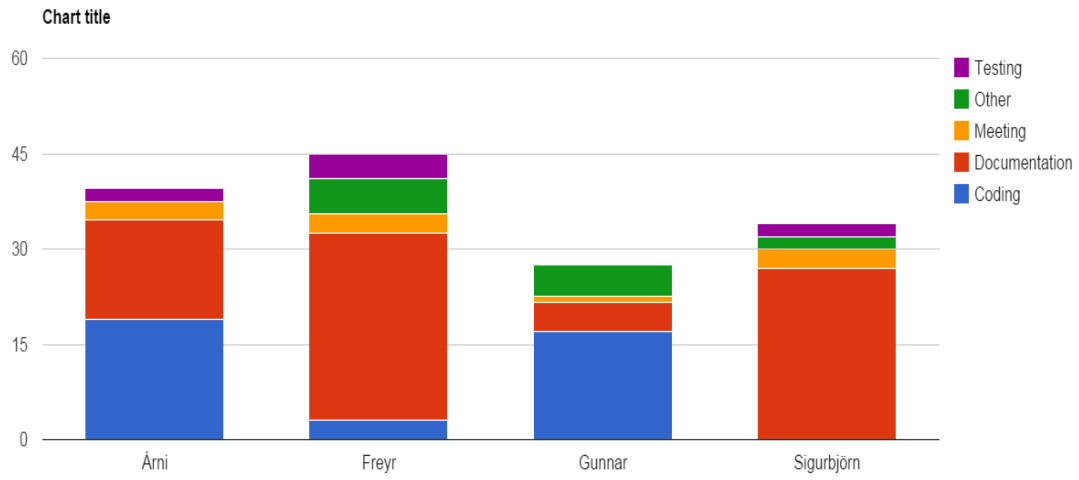
What went well

- We got something to work (which shows some functionality)

What could be done better

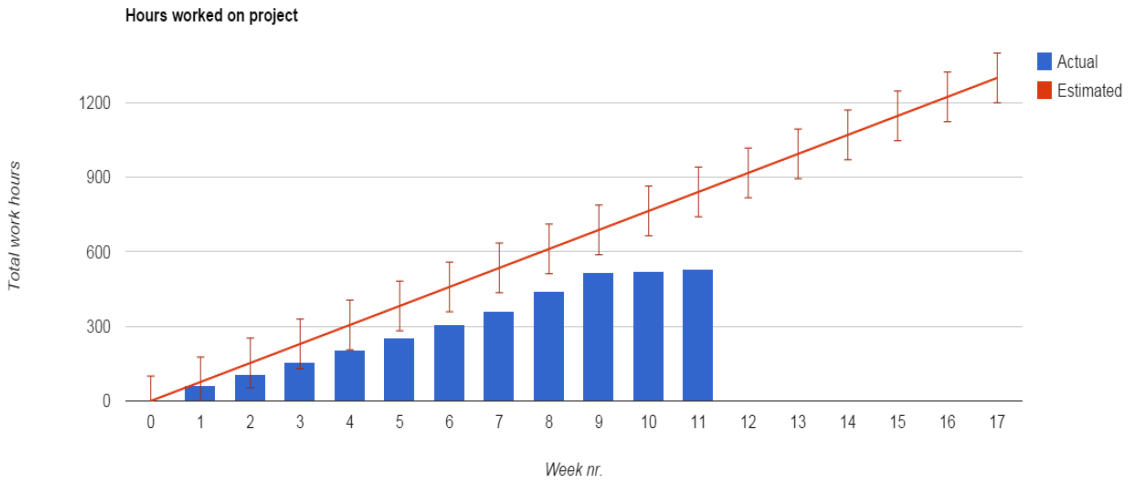
- Attendance at daily meetings

Time registration for sprint 4



Sprint 5 (Jazz)

Burn-up chart (updated March 29)

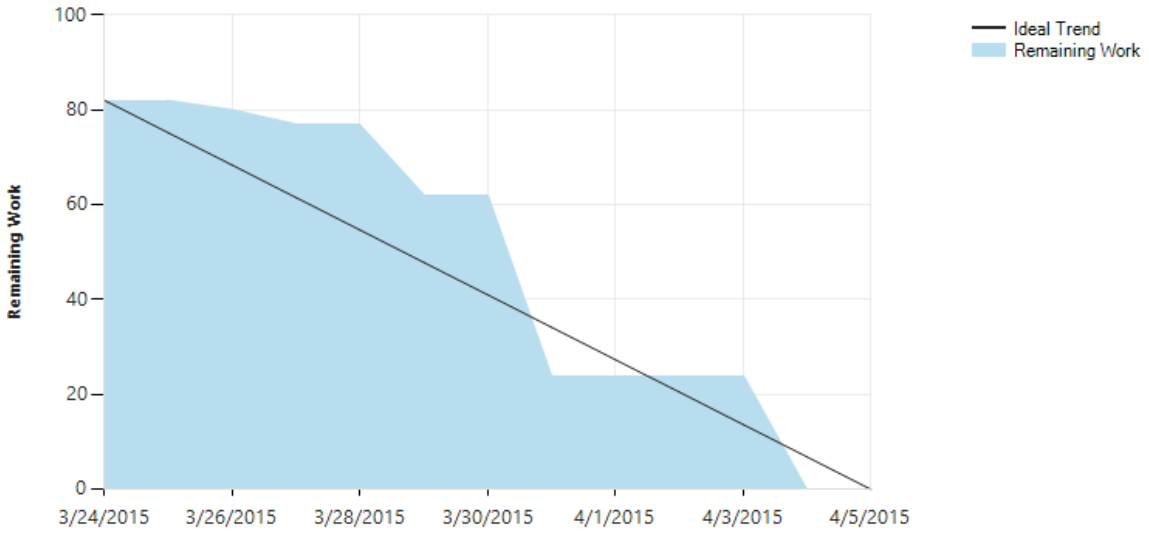


Sprint planning

Sprint Backlog for Sprint 5	Est.
As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	1 sp
• Update Development Report (Framvinduskýrsla) after sprint 4	2 hrs
• Update Project Schedule (Verkáætlun) after Sprint 4	2 hrs
As an instructor and examiner I want to see the project documents updated with regards to points made in Status Meeting 2	1 sp
• Create a document that shows the status of the project, which scripts are finished, etc.	4 hrs
As a user I want to be able to fetch a build from ParaBuild so that I can include it in the delivery	5 sp
• Retrieve the URL for the Parabuild server from Config Manager	2 hrs
• Retrieve option whether to include source tarballs from the Config Manager	2 hrs
• Retrieve the Staging Directory from the Config Manager	1 hr

• Retrieve URL for build tarballs from Parabuild using the SOAP API	20 hrs
• Update the Delivery Description field in the Delivery object after tarballs have been retrieved	12 hrs
• Retrieve build tarballs from Parabuild and save them to the staging directory	6 hrs
• Create unit test for the script	7 hrs
As a user I want the delivery to show up in a list of deliveries (in Django) so I can see what has been delivered	2 sp
• Create a view in Django	6 hrs
• Show a list of deliveries in the view	2 hrs
As a user I want the delivery output files to be uploaded to a storage server so that I have a backup of the data	3 sp
• Retrieve hostname, username and password to the storage server from Config Manager	2 hrs
• Retrieve the Delivery Directory from the Config Manager	1 hr
• Upload files in the delivery directory to the storage host using the server credentials	6 hrs
• Create a unit test for the script	7 hrs

Sprint burndown



Team's velocity

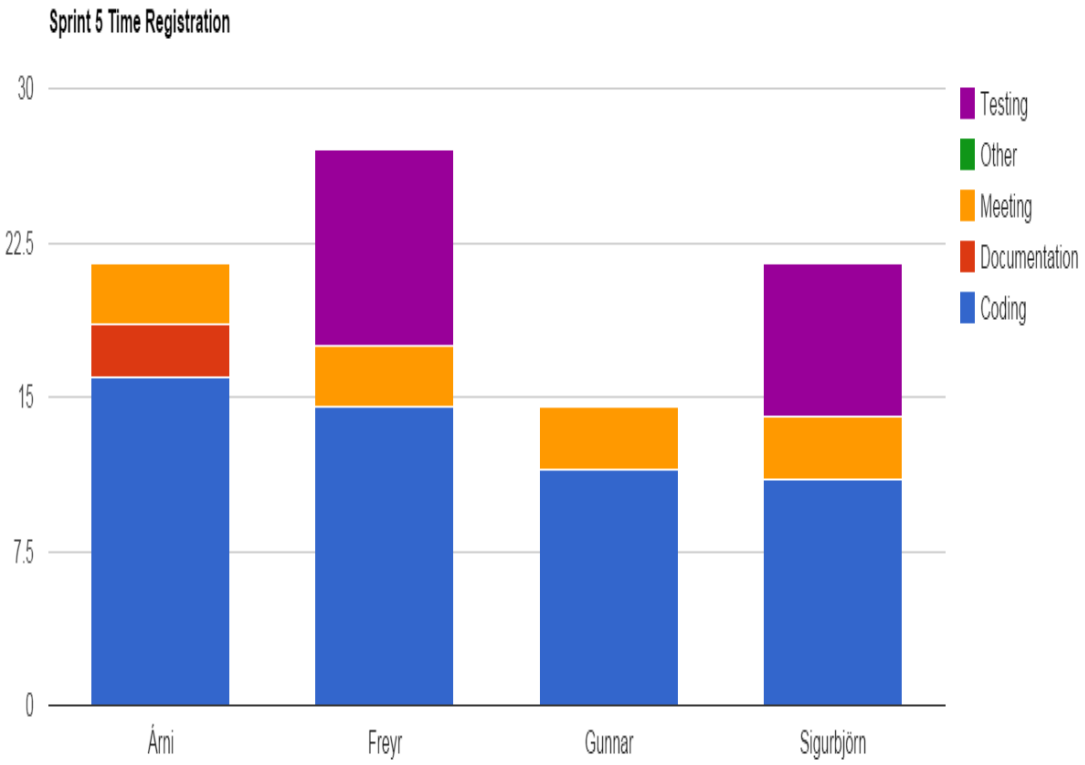
The velocity for sprint 5 was 12 story points

Sprint 5 retrospective

Actions

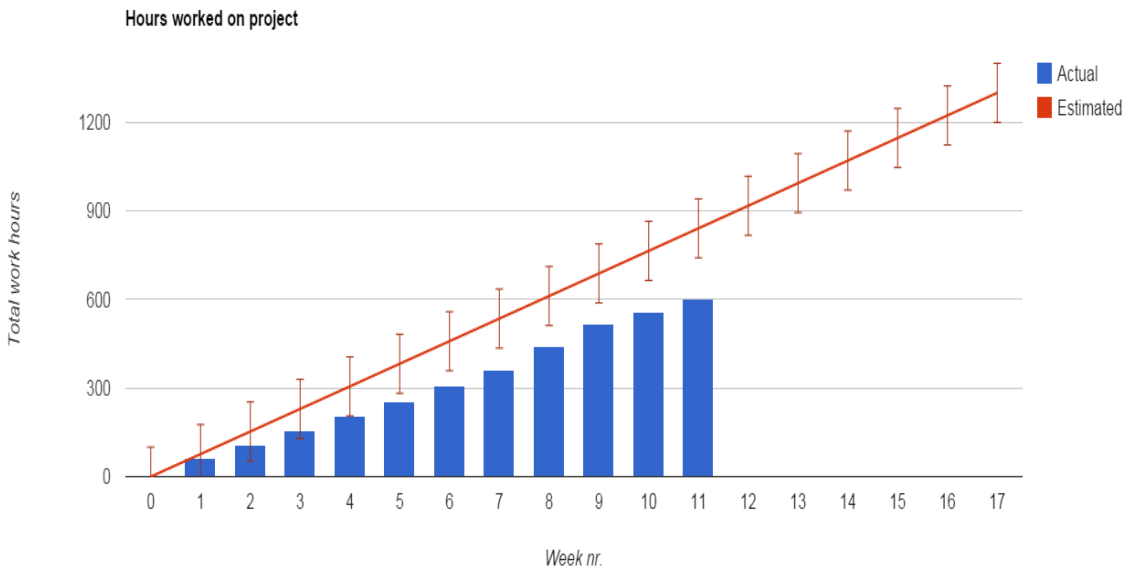
- check timeout for larger files when copying to storage server
- add to Risk Analysis (Áhættugreining) risk of cloud based programs
- explain to instructors about low priority of web front end

Time registration for sprint 5



Sprint 6 (Optimus Prime), April 8th - April 19th

Burn-up chart (updated April 8th)

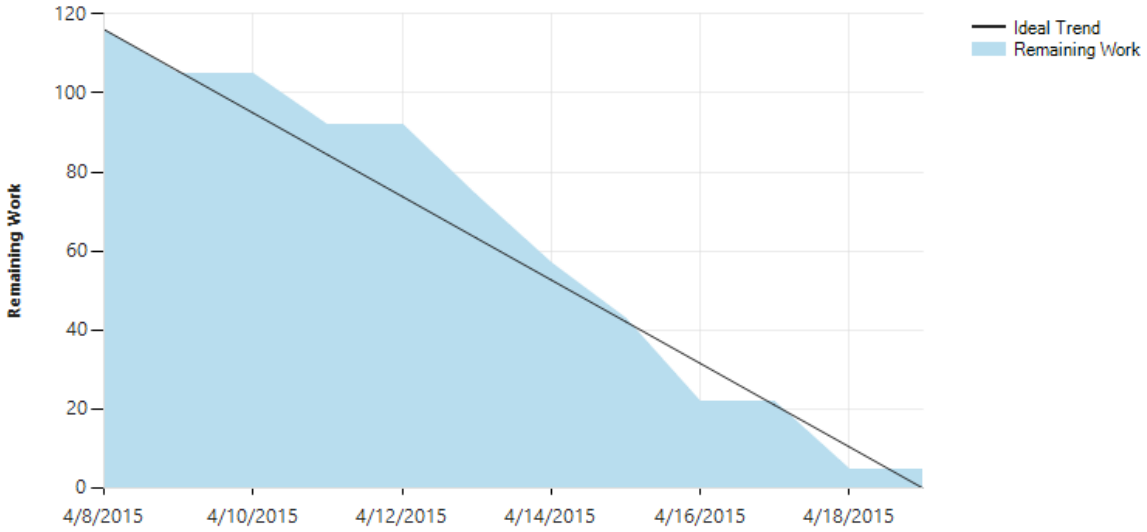


Sprint planning

Sprint Backlog for Sprint 6	Est.
As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	1 sp
• Update Development Report (Framvinduskýrsla) after sprint 4	2 hrs
• Update Project Schedule (Verkáætlun) after Sprint 4	2 hrs
As a developer I want a parsing tool to recognize task headers so that it becomes easier to examine them in scripts	5 sp
• Design exactly how the parser should work	20 hrs
• Implement interface for the FileParser module	1 hr
• Create unit tests for the parse function in FileParser	13 hrs
• Get actual task headers from Tern code base to use for unit testing	8 hrs
• Create RegEx for first line of Task Header	3 hrs

• Implement parse_date function	1 hr
• Have parse_date return actual date parsed, or None if date field is unparsable	13 hrs
• Create unit tests for the parse_date function in the FileParser module	13 hrs
As a developer I want to connect to TargetProcess through an API to get access to task numbers	4 sp
• Investigate TargetProcess API	10 hrs
• Connect to Tern's TargetProcess site with credentials from the Config Manager	4 hrs
• Add TargetProcess credentials to Config Manager	1 hr
• Get information about a specific Product Release from TargetProcess, using the ProductVersion of a delivery	5 hrs
• Get list of task numbers for the given product version	4 hrs
As a user I want to make sure that task header numbers are correctly numbered in an incremental fashion so that the delivery isn't rejected	2 sp
• Get a list of new and modified files since the previous release	10 hrs
• Parse the new and modified files with the File Parser	5 hrs
• Check validity of task headers entry numbers of task headers returned from File Parser	5 hrs
• Investigate how to get a list of new and modified files from Perforce	2 hrs
• Investigate how to get the contents of a file from Perforce	2 hrs
• Investigate how to get the difference of a file from previous delivery	2 hrs

Sprint burndown



(a completed task's status was not set as done, therefore the chart shows unfinished work)

Team velocity

The velocity for sprint 6 was 12 story points.

Sprint 6 retrospective

What went well

- Work on scripts went well

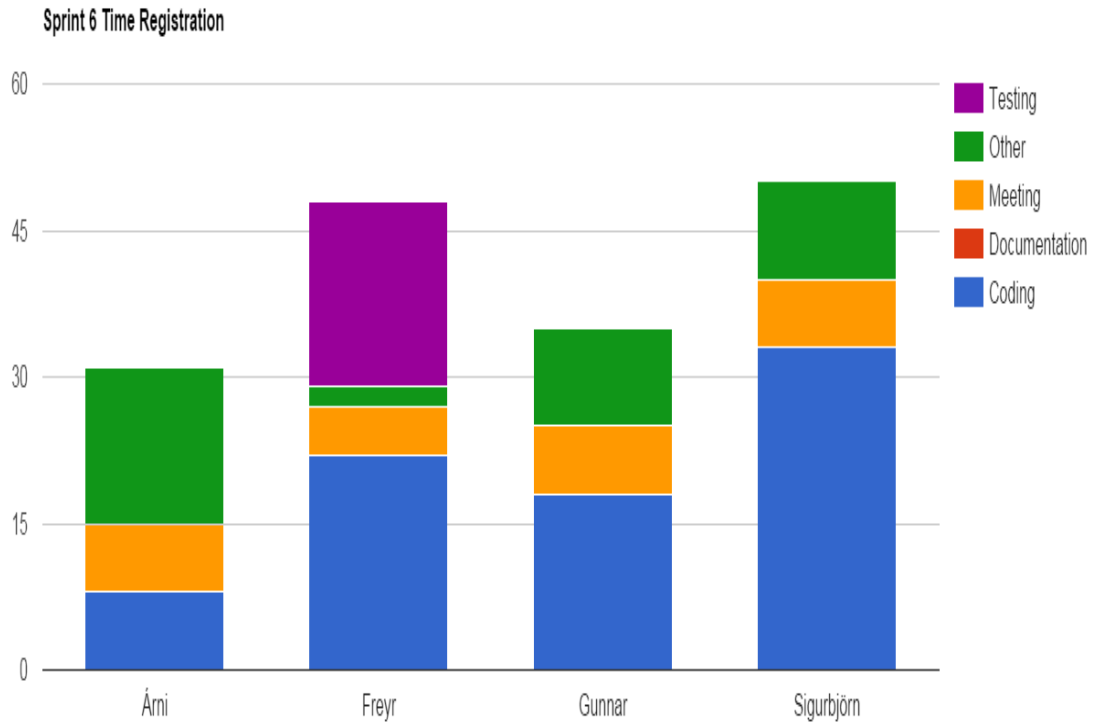
What could be done better

- Less capacity because of exams was not taken into account in sprint planning

Actions

- Update copy_delivery_to_storage_server so it does not overwrite existing
- Sharepoint config - folder instead of url

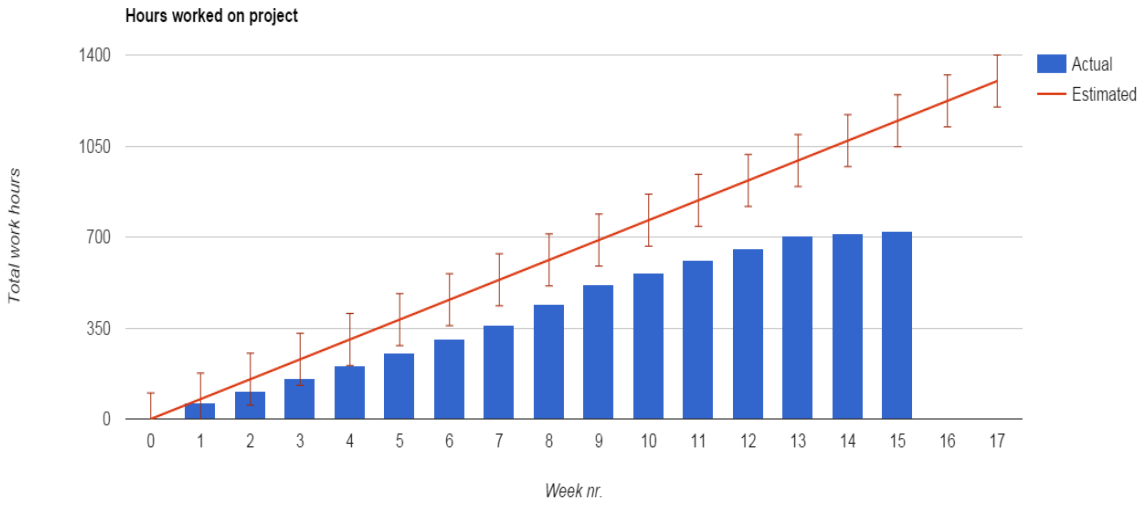
Time registration for sprint 6



April 21st - April 26th: 6 days time off due to exams and family obligations

Sprint 7 (Hot Rod), April 27th - May 3rd

Burn-up chart (updated April 26th)



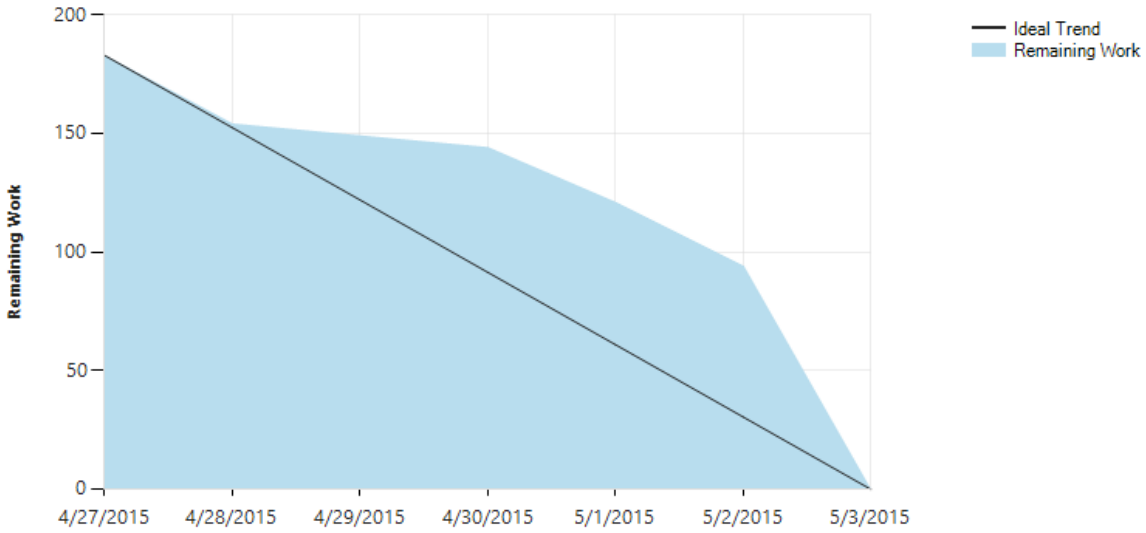
Sprint planning

Sprint Backlog for Sprint 7	Est.
As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	1 sp
• Update Development Report (Framvinduskýrsla) after sprint 4	4 hrs
• Update Project Schedule (Verkáætlun) after Sprint 4	3 hrs
• Update risk analysis because of risk using cloud based service	4 hrs
As a user I want to make sure that the task headers delivered contain valid task numbers so that the delivery isn't rejected	3 sp
• Get task numbers in current delivery from the Delivery object	5 hrs
• Get list of new and modified files in this delivery compared to last delivery	2 hrs

• Parse task headers in new and modified files	2 hrs
• Go through all new task header entries and compare to list of task numbers in delivery	15 hrs
• Make sure the task numbers in the task header entries are task numbers in the delivery	2 hrs
• Make sure each task number in the delivery is mentioned at least once in the task header entries	15 hrs
As a user I want task headers to show the correct date in task headers so that the delivery isn't rejected	5 sp
• Get list of new and modified files in this delivery compared to last delivery	2 hrs
• Parse task headers in new and modified files	2 hrs
• Go through all new task header entries and compare date format to date format in previous task header entry	15 hrs
• Make sure dates in new task header entries are not in the future	8 hrs
• Make sure the date in new task header entries are not older than the date of the previous delivery	8 hrs
As a user I want to retrieve all documents from a folder in SharePoint for a product instead of one document	3 sp
• Have a config value for the directory of documents for each product with a placeholder for the product version	2 hrs
• Update script to iterate through folder instead of using url for one file and download each file in the folder	20 hrs
As a user I don't want old backup files to be overwritten in case of mistakes in new delivery	3 sp
• Investigate transactions of large files when copying to storage server	5 hrs
• Rename existing folder to a unique name before uploading delivery directory	15 hrs
As an instructor and examiner I want to see a presentation of the product for the third status meeting (Stöðufundur 3) so I can assess the progress of the project	1 sp
• Update operational manual, and user guide	10 hrs

• Review other documents and update accordingly	20 hrs
• Create a presentation for Status Meeting 3	20 hrs
• Rehearse presentation	20 hrs
As a user I want to be able to create a new delivery with a command line script or perform re-delivery on an existing delivery	3 sp
• Update create delivery script	15 hrs

Sprint burndown



Team Velocity

The velocity for sprint 7 was 19 story points.

Sprint 7 retrospective

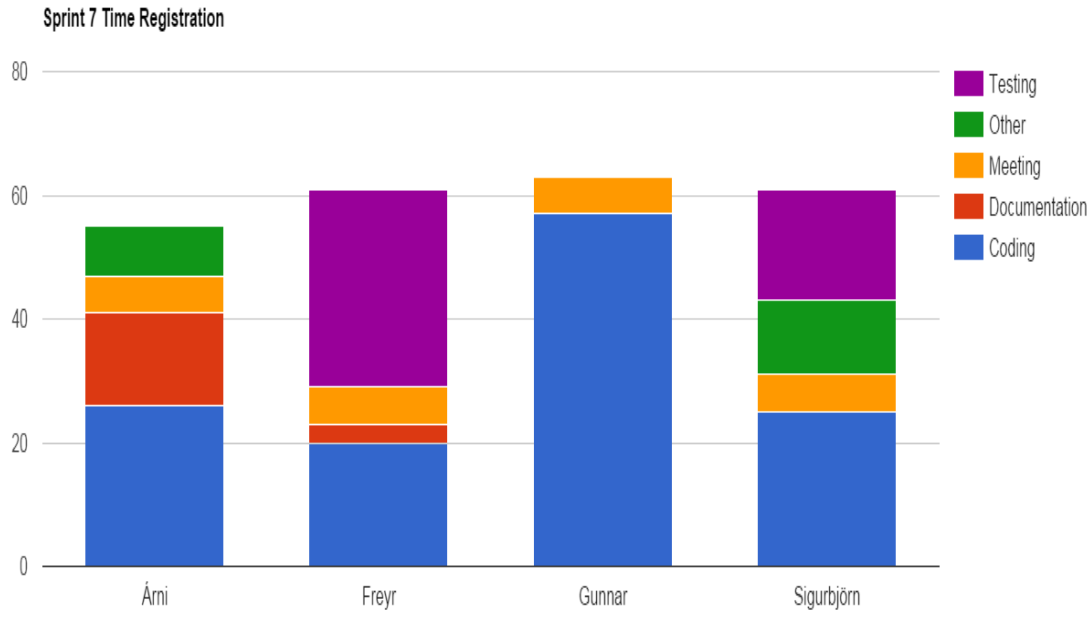
What could be done better

- Unit tests were challenging

Actions

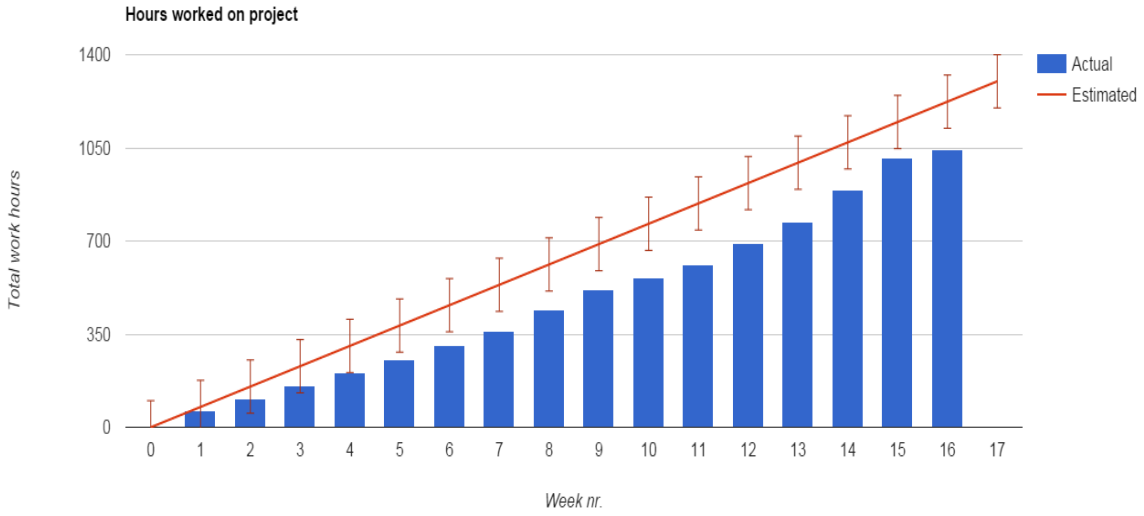
- Make use of collaboration when coding

Time registration for sprint 7



Sprint 8 (Mirage), May 4th - May 10th

Burn-up chart (updated May 4th)



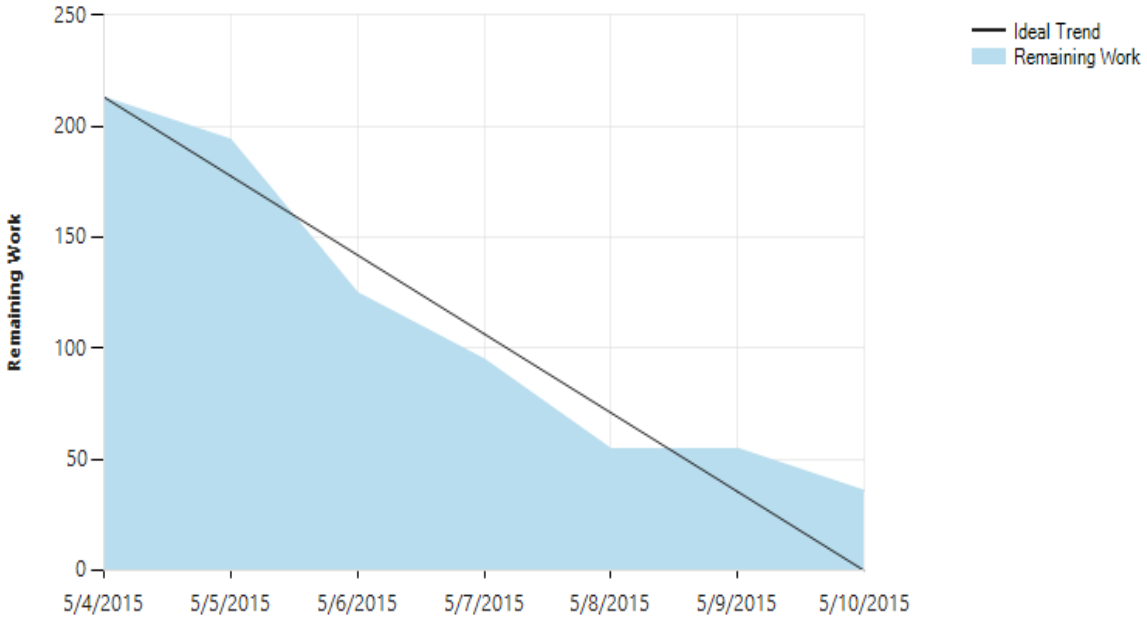
Sprint planning

Sprint Backlog for Sprint 8	Est.
As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	1 sp
• Update Development Report (Framvinduskýrsla) after sprint 4	4 hrs
• Update Project Schedule (Verkáætlun) after Sprint 4	3 hrs
Email all documents to instructor and examiner	1 hr
As a user I want to make sure that the task headers delivered contain valid task numbers so that the delivery isn't rejected	3 sp
• Make sure the task numbers in the task header entries are task numbers in the delivery	2 hrs
• Write unit test for the script	15 hrs

As an instructor and examiner I want to see a presentation of the product for the third status meeting (Stöðufundur 3) so I can assess the progress of the project	1 sp
● Create a presentation for Status Meeting 3	20 hrs
● Rehearse presentation	20 hrs
Do a retrospective meeting after Status Meeting 3	1 hrs
As a user I want to view one specific delivery report so I can verify its correctness	3 sp
Create a view in Django for viewing of single delivery reports	25 hrs
As a user I want to generate a delivery report in PDF format so that I can add it to the delivery	3 sp
Install and investigate how to use PDFKit	2 hrs
Create PDF from view of a specific delivery report	10 hrs
Store PDF of delivery document in Delivery Directory	5 hrs
As an instructor and examiner I want to see the project documents updated with regards to points made in Status Meeting 3	1 sp
Placeholder for things to do with regards to points made in status meeting 3	20 hrs
As a user I want the run_script script to behave the same as the Script.execute method	1 sp
Update run_script.py script	5 hrs
As a user I don't need config values for a specific product version (only for products)	1 sp
Remove ProductVersion from Config model	5 hrs
As a user I want the system to work on product versions regardless of minor version being zero padded or not	3 sp
Update retrieve_document_from_sharepoint to use zero-padded minor version	10 hrs
Investigate the need to update other scripts w.r.t. zero padded minor version	10 hrs
As a user I want the system to automatically get a list of new, modified and removed files between to deliveries	4 sp
Create unit test for script	15 hrs

Read data from perforce to create files	15 hrs
Store files in staging directory	2 hrs
As a user I want the system to automatically create an ISO of a staging directory and MD5	3 sp
Create unit tests for script	15 hrs
Create an ISO image of the staging directory	10 hrs
Store ISO image and MD5 checksum of it in the delivery directory	5 hrs

Sprint burndown



(Two tasks were left unfinished in the sprint)

Team velocity

The velocity for Sprint 8 was 23 story points

Retrospective after Status Meeting 3

Actions

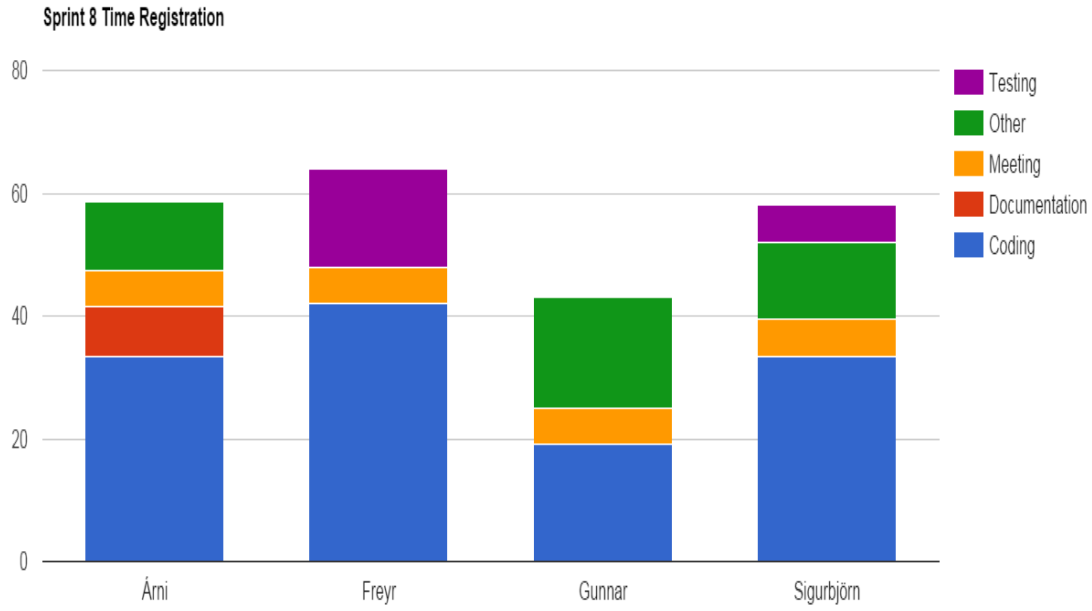
- Deployment of DATI
  - from runserver(Django) to Apache server
- Demo failure
  - new delivery does not work - missing tern_base_tag and tern_delivery_tag
- For final presentation:

    - show the sequence diagram for each script when running demo
- Finish info log for all scripts

Sprint review

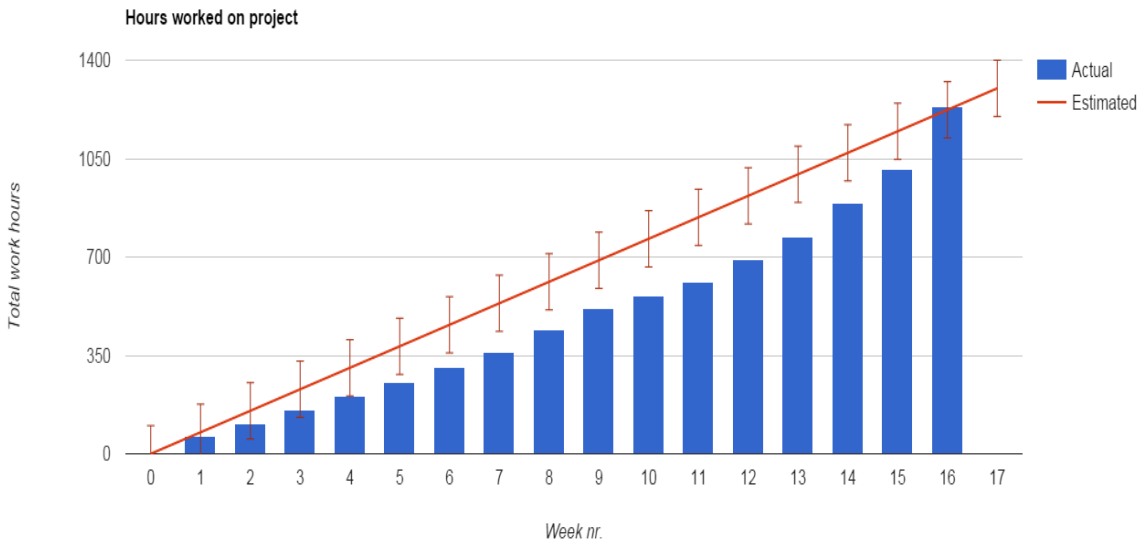
TODO

Time registration



Sprint 9 (Mirage), May 11th - May 18th

Burn-up chart (updated May 11th)

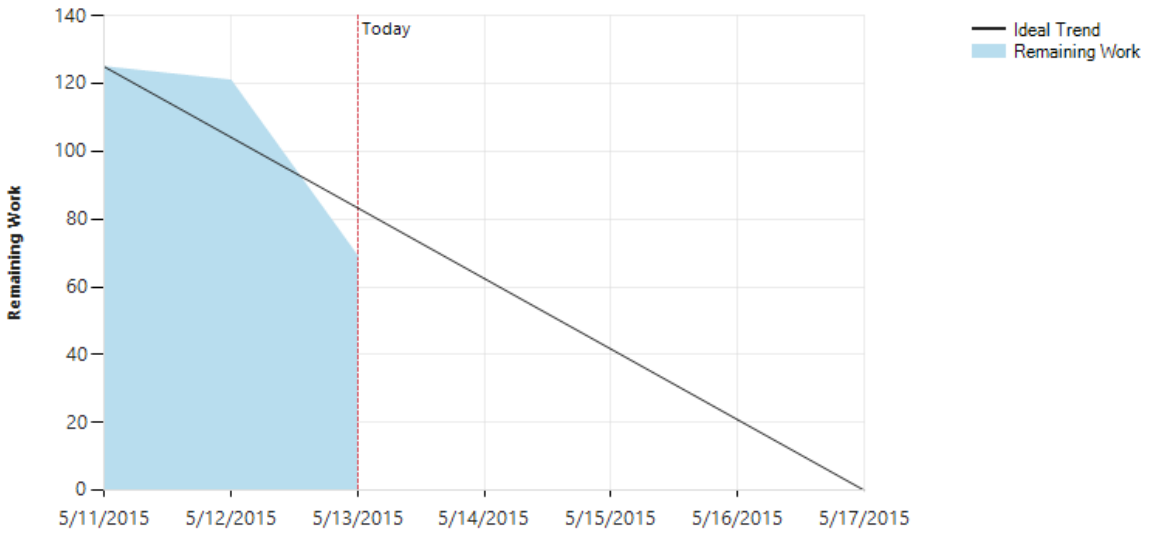


Sprint planning

Sprint Backlog for Sprint 9	Est.
As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	1 sp
• Update Development Report (Framvinduskýrsla) after sprint 4	4 hrs
• Update Project Schedule (Verkáætlun) after Sprint 4	3 hrs
As an instructor and examiner I want to see the project documents updated with regards to points made in Status Meeting 3	1 sp
• Update operational manual to include deployment instructions	6 hrs
• Investigate how to perform deployment of DATI to production server	20 hrs
As an instructor and examiner I want to see a final report for the project so I can grade the project	8 sp
• Finish the introduction section of the final report	10 hrs
• Finish the working arrangements of the final report	10 hrs

• Finish the progress section of the final report	10 hrs
• Finish the conclusion section of the final report	10 hrs
• Finish the appendices of the final report	10 hrs
• Send the report to the printer BEFORE wednesday 15:00	1 hr
As an instructor and examiner I want to see a presentation of the product for the final presentation	3 sp
• Create a presentation	20 hrs
• Rehearse presentation	20 hrs

Sprint burndown



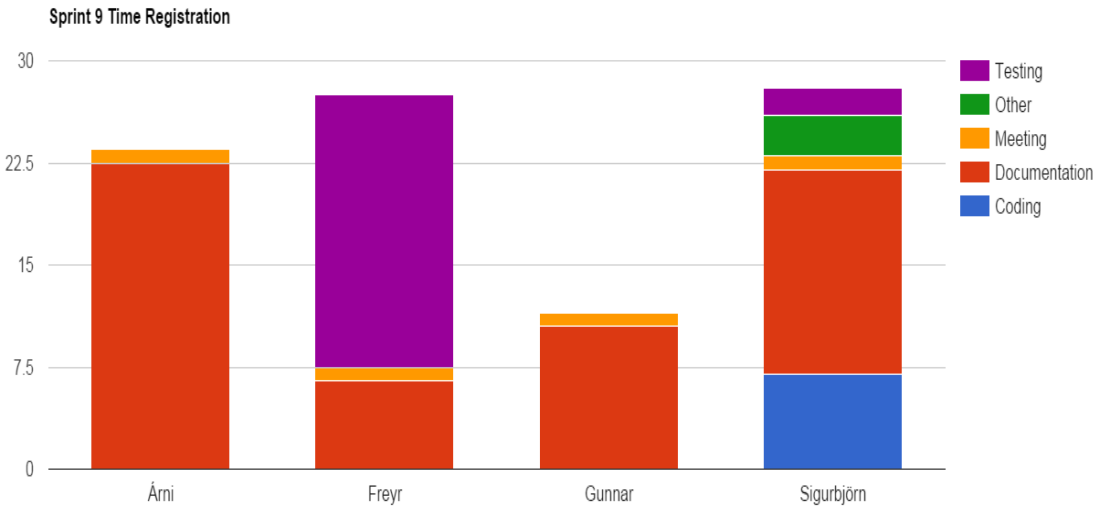
Team velocity

N/A as sprint is not finished

Sprint review

N/A

Time registration



Tern Systems - Work journal (Dagbók)

T-404-LOKA

Students:

Árni Þorvaldsson (arnitho12@ru.is)

Freyr Bergsteinsson (freyrb12@ru.is)

Gunnar Þór Helgason (gunnarh07@ru.is)

Sigurbjörn Kristjánsson (sigurbjorn12@ru.is)

Instructor:

Daníel Máni Jónsson (daniel@valitor.is)

Examiner:

Hannes Pétursson (hap@ru.is)

Date	Who	Hours (1381)	Type	Description of what was done
2015-01-13	Árni	1	Meeting	Lecture at RU, Hallgrímur Arnalds
2015-01-13	Freyr	1	Meeting	Lecture at RU, Hallgrímur Arnalds
2015-01-13	Gunnar	1	Meeting	Lecture at RU, Hallgrímur Arnalds
2015-01-13	Sigurbjörn	1	Meeting	Lecture at RU, Hallgrímur Arnalds
2015-01-14	Árni	3	Meeting	Project introduction and first draft of project report
2015-01-14	Freyr	3	Meeting	Project introduction and first draft of project report
2015-01-14	Gunnar	3	Meeting	Project introduction and first draft of project report
2015-01-14	Sigurbjörn	3	Meeting	Project introduction and first draft of project report
2015-01-16	Árni	0.5	Meeting	Meeting with Tern's correspondent (Heiðar)
2015-01-16	Freyr	0.5	Meeting	Meeting with Tern's correspondent (Heiðar)
2015-01-16	Gunnar	0.5	Meeting	Meeting with Tern's correspondent (Heiðar)
2015-01-16	Sigurbjörn	0.5	Meeting	Meeting with Tern's correspondent (Heiðar)
2015-01-18	Freyr	0.5	Other	Updated project report with tips from instructor
2015-01-19	Gunnar	4.5	Documentation	Planning sprint 0, Trello board, updating project report
2015-01-19	Árni	4.5	Documentation	Planning sprint 0, updating project report
2015-01-19	Freyr	4.5	Documentation	Planning sprint 0, updating project report
2015-01-19	Sigurbjörn	4.5	Documentation	Planning sprint 0, updating project report
2015-01-20	Freyr	1	Meeting	Lecture at RU, Hannes Pétursson
2015-01-20	Sigurbjörn	1	Meeting	Lecture at RU, Hannes Pétursson
2015-01-20	Árni	1	Meeting	Lecture at RU, Hannes Pétursson
2015-01-20	Gunnar	1	Meeting	Lecture at RU, Hannes Pétursson
2015-01-20	Freyr	4	Documentation	Sprint 0, updating project report
2015-01-20	Sigurbjörn	4	Documentation	Sprint 0, updating project report
2015-01-20	Árni	4	Documentation	Sprint 0, updating project report
2015-01-20	Gunnar	4	Documentation	Sprint 0, updating project report
2015-01-24	Freyr	4	Other	Set up Visual Studio Online account and set up backlog and sprints
2015-01-25	Árni	2	Other	Django installation, github repository
2015-01-26	Árni	4.5	Meeting	Sprint 0 review, Retrospective, Poker planning and Sprint 1 planning
2015-01-26	Freyr	4.5	Meeting	Sprint 0 review, Retrospective, Poker planning and Sprint 1 planning
2015-01-26	Gunnar	4.5	Meeting	Sprint 0 review, Retrospective, Poker planning and Sprint 1 planning
2015-01-26	Sigurbjörn	4.5	Meeting	Sprint 0 review, Retrospective, Poker planning and Sprint 1 planning
2015-01-27	Árni	1	Meeting	Lecture at RU, Torfi Leifsson
2015-01-27	Freyr	1	Meeting	Lecture at RU, Torfi Leifsson
2015-01-27	Gunnar	1	Meeting	Lecture at RU, Torfi Leifsson
2015-01-27	Sigurbjörn	1	Meeting	Lecture at RU, Torfi Leifsson
2015-01-27	Árni	5	Documentation	Sprint 1, Project Schedule documentation
2015-01-27	Freyr	5	Documentation	Sprint 1, Working Arrangements and Project Schedule documentation
2015-01-27	Gunnar	5	Documentation	Sprint 1, Working Arrangements and Project Schedule documentation
2015-01-27	Sigurbjörn	5	Documentation	Sprint 1, Project Schedule documentation
2015-01-29	Árni	2.5	Documentation	Sprint 1, Project Presentation (5 min)
2015-01-29	Freyr	2.5	Documentation	Sprint 1, Project Presentation (5 min)
2015-01-29	Gunnar	2.5	Documentation	Sprint 1, Project Presentation (5 min)
2015-01-29	Sigurbjörn	2.5	Documentation	Sprint 1, Project Presentation (5 min)
2015-02-02	Gunnar	5	Coding	Sprint 1, Creating test Config manual and creating config manual
2015-02-02	Freyr	5	Coding	Sprint 1, Setting up development machine, config manager
2015-02-02	Árni	5	Coding	Sprint 1, frontend html, master script
2015-02-02	Sigurbjörn	5	Coding	Sprint 1, Setting up development machine, config manager
2015-02-03	Árni	2	Meeting	Student presentations at RU
2015-02-03	Freyr	2	Meeting	Student presentations at RU
2015-02-03	Gunnar	2	Meeting	Student presentations at RU
2015-02-03	Sigurbjörn	2	Meeting	Student presentations at RU
2015-02-05	Árni	1.5	Coding	Sprint 1, master script
2015-02-05	Freyr	2	Testing	Sprint 1, Creating tests for Config manager
2015-02-05	Gunnar	2	Testing	Sprint 1, Creating tests for Config manager
2015-02-05	Sigurbjörn	2	Coding	Sprint 1, master script
2015-02-07	Árni	1.5	Coding	Sprint 1, master script
2015-02-07	Freyr	1.5	Coding	Sprint 1, Creating test Config manual and creating config manual
2015-02-07	Gunnar	1.5	Coding	Sprint 1, Creating test Config manual and creating config manual
2015-02-09	Árni	5	Meeting	Sprint 1 review, Retrospective and Sprint 2 planning
2015-02-09	Freyr	5.5	Meeting	Sprint 1 review, Retrospective and Sprint 2 planning
2015-02-09	Gunnar	5.5	Meeting	Sprint 1 review, Retrospective and Sprint 2 planning

Date	Who	Hours (1381)	Type	Description of what was done
2015-02-09	Sigurbjörn	5.5	Meeting	Sprint 1 review, Retrospective and Sprint 2 planning
2015-02-10	Árni	2.5	Documentation	Sprint 2, Development Report (framvinduskýrsla)
2015-02-10	Freyr	2.5	Documentation	Sprint 2, Sequence diagram of the master script
2015-02-10	Gunnar	2.5	Documentation	Sprint 2, Risk Analysis (Áhættuskýrsla)
2015-02-12	Árni	3	Documentation	Sprint 2, Development Report
2015-02-12	Freyr	3	Documentation	Sprint 2, Update system architecture and add sequence diagram to project report
2015-02-12	Gunnar	3	Documentation	Sprint 2, Various documentation
2015-02-12	Sigurbjörn	3	Documentation	Sprint 2, Various documentation
2015-02-15	Árni	3	Documentation	Sprint 2, Development Report
2015-02-15	Freyr	3	Documentation	Sprint 2, Development Report, Presentation for first status meeting
2015-02-15	Gunnar	3	Documentation	Sprint 2, Risk Analysis (Áhættuskýrsla)
2015-02-15	Sigurbjörn	3	Documentation	Sprint 2, Risk Analysis, Presentation
2015-02-15	Árni	0.5	Meeting	Sprint 2, Meeting with Instructor
2015-02-15	Freyr	0.5	Meeting	Sprint 2, Meeting with Instructor
2015-02-15	Gunnar	0.5	Meeting	Sprint 2, Meeting with Instructor
2015-02-15	Sigurbjörn	0.5	Meeting	Sprint 2, Meeting with Instructor
2015-02-16	Árni	4	Meeting	Sprint 2, Preparation for first status meeting presentation
2015-02-16	Freyr	4	Meeting	Sprint 2, Preparation for first status meeting presentation
2015-02-16	Gunnar	3.5	Meeting	Sprint 2, Preparation for first status meeting presentation
2015-02-16	Sigurbjörn	4	Meeting	Sprint 2, Preparation for first status meeting presentation
2015-02-17	Árni	1	Meeting	Sprint 2, First status meeting
2015-02-17	Freyr	1	Meeting	Sprint 2, First status meeting
2015-02-17	Gunnar	1	Meeting	Sprint 2, First status meeting
2015-02-17	Sigurbjörn	1	Meeting	Sprint 2, First status meeting
2015-02-21	Sigurbjörn	4	Coding	Sprint 2, Work on coding the master script
2015-02-21	Sigurbjörn	1	Meeting	Sprint 2, Meeting with Instructor
2015-02-21	Freyr	1	Meeting	Sprint 2, Meeting with Instructor
2015-02-21	Árni	1	Meeting	Sprint 2, Meeting with Instructor
2015-02-21	Gunnar	1	Meeting	Sprint 2, Meeting with Instructor
2015-02-21	Freyr	5	Coding	Sprint 2, Database schema for config manager and delivery reports
2015-02-21	Árni	4	Coding	Sprint 2, Work on coding the master script
2015-02-21	Gunnar	5	Coding	Sprint 2, Database schema for config manager and delivery reports
2015-02-22	Sigurbjörn	2	Coding	Sprint 2, working on error handling in master script
2015-02-22	Árni	2	Coding	Sprint 2, working on error handling in master script
2015-02-23	Sigurbjörn	4	Meeting	Sprint 2 review, Retrospective and Sprint 3 planning
2015-02-23	Freyr	4	Meeting	Sprint 2 review, Retrospective and Sprint 3 planning
2015-02-23	Árni	4	Meeting	Sprint 2 review, Retrospective and Sprint 3 planning
2015-02-23	Gunnar	4	Meeting	Sprint 2 review, Retrospective and Sprint 3 planning
2015-02-24	Sigurbjörn	2.5	Documentation	Updated risk analysis with tips from status meeting 1
2015-02-24	Freyr	3	Testing	Investigating unit tests with Django
2015-02-24	Árni	3	Other	Learning about Django
2015-02-24	Gunnar	2	Testing	Investigating unit tests with Django
2015-02-28	Sigurbjörn	5	Coding	Investigating delivery report generation
2015-02-28	Freyr	6	Testing	Implementing config manager unit tests
2015-02-28	Gunnar	5.5	Testing	Implementing config manager unit tests
2015-03-02	Sigurbjörn	5	Coding	Implement createNewDelivery function in the config manager
2015-03-02	Freyr	5.5	Testing	Implementing config manager and product unit tests
2015-03-02	Gunnar	5.5	Testing	Implementing config manager and product unit tests
2015-03-02	Árni	5	Coding	Implement createNewDelivery function in the config manager
2015-03-03	Sigurbjörn	3.5	Testing	Implement config manager unit tests
2015-03-03	Freyr	3.5	Coding	Database schema
2015-03-03	Gunnar	2	Testing	Implement config manager unit tests
2015-03-03	Árni	3.5	Testing	Implement config manager unit tests
2015-03-07	Sigurbjörn	6	Testing	Implement unit tests for delivery model
2015-03-07	Freyr	6.5	Documentation	Design of scripts, documentation and sequence diagrams
2015-03-07	Gunnar	6.5	Documentation	Sequence Diagrams
2015-03-07	Árni	5	Testing	Implement unit tests for delivery model
2015-03-07	Freyr	1	Coding	Refactoring of models
2015-03-08	Árni	3	Testing	Implement unit tests for delivery model
2015-03-08	Sigurbjörn	3	Testing	Implement unit tests for delivery model
2015-03-09	Árni	3	Meeting	Sprint 3 review, Retrospective and Sprint 4 planning

Date	Who	Hours (1381)	Type	Description of what was done
2015-03-09	Freyr	3	Meeting	Sprint 3 review, Retrospective and Sprint 4 planning
2015-03-09	Sigurbjörn	3	Meeting	Sprint 3 review, Retrospective and Sprint 4 planning
2015-03-09	Árni	2	Testing	Unit tests for delivery model
2015-03-09	Freyr	2	Documentation	Updating design document with diagrams for scripts
2015-03-09	Sigurbjörn	2	Testing	Unit tests for delivery model
2015-03-09	Gunnar	1	Meeting	Sprint 3 review, Retrospective and Sprint 4 planning (went home sick)
2015-03-11	Freyr	2	Documentation	Updating design document with diagrams for scripts
2015-03-10	Árni	4	Documentation	Updating development report, project report, working arrangements
2015-03-10	Freyr	4	Documentation	
2015-03-10	Sigurbjörn	4	Documentation	
2015-03-12	Árni	2	Documentation	Updating risk analysis, project schedule
2015-03-14	Sigurbjörn	4.5	Documentation	Presentation for 2nd Status Meeting
2015-03-14	Gunnar	4.5	Documentation	diagrams
2015-03-14	Freyr	4.5	Documentation	Sequence diagrams for the Design Document
2015-03-14	Árni	4.5	Documentation	Updating project schedule, work journal,
2015-03-16	Gunnar	5.5	Coding	Updating masterscript,
2015-03-16	Freyr	5.5	Documentation	Sequence diagrams for the Design Document
2015-03-16	Sigurbjörn	5	Documentation	Presentation for 2nd Status Meeting
2015-03-16	Árni	5.5	Coding	Updating masterscript, added models to the admin interface in django
2015-03-17	Gunnar	3.5	Coding	update_product_versions script
2015-03-17	Freyr	4	Testing	update_product_versions script and Script/ProductScript unit tests
2015-03-17	Árni	4	Coding	update_product_versions script
2015-03-17	Freyr	1	Coding	Implementation of Script/ProductScript models
2015-03-21	Sigurbjörn	8.5	Documentation	Presentation for 2nd Status Meeting
2015-03-21	Gunnar	8	Coding	update_product_versions
2015-03-21	Freyr	8.5	Documentation	Operations Manual and User Guide drafts
2015-03-21	Árni	7	Coding	update_product_versions
2015-03-22	Árni	2.5	Coding	
2015-03-22	Freyr	5.5	Other	Investigating logging in to SharePoint through REST
2015-03-22	Sigurbjörn	2	Other	
2015-03-23	Gunnar	5	Other	
2015-03-23	Sigurbjörn	5	Documentation	
2015-03-23	Árni	5	Documentation	
2015-03-23	Freyr	2	Coding	Logging in to SharePoint through REST
2015-03-23	Freyr	3	Documentation	Presentation for 2nd Status Meeting
2015-03-24	Gunnar	2	Meeting	Status Meeting 2 @ Tern and Status Meeting Retrospective
2015-03-24	Sigurbjörn	2	Meeting	Status Meeting 2 @ Tern and Status Meeting Retrospective
2015-03-24	Árni	2	Meeting	Status Meeting 2 @ Tern and Status Meeting Retrospective
2015-03-24	Freyr	2	Meeting	Status Meeting 2 @ Tern and Status Meeting Retrospective
2015-03-24	Gunnar	3	Meeting	Sprint 4 review, Retrospective and Sprint 5 planning
2015-03-24	Sigurbjörn	3	Meeting	Sprint 4 review, Retrospective and Sprint 5 planning
2015-03-24	Árni	3	Meeting	Sprint 4 review, Retrospective and Sprint 5 planning
2015-03-24	Freyr	3	Meeting	Sprint 4 review, Retrospective and Sprint 5 planning
2015-03-26	Freyr	2	Coding	Retrieve build from parabuild
2015-03-29	Freyr	7.5	Testing	Script unit testing
2015-03-29	Gunnar	7.5	Coding	script document and Django view
2015-03-29	Sigurbjörn	7.5	Testing	Script unit testing
2015-03-29	Árni	2.5	Documentation	Development Report (Framvinduskýrsla), Project Schedule (Verkáætlun)
2015-03-29	Árni	3.5	Coding	
2015-03-30	Freyr	4	Coding	Retrieve build from parabuild
2015-03-30	Gunnar	4	Coding	Django view for deliveries
2015-03-30	Sigurbjörn	4	Coding	Retrieve build from parabuild
2015-03-30	Árni	4	Coding	Django view for deliveries
2015-03-31	Freyr	3.5	Coding	Retrieve build from parabuild
2015-03-31	Gunnar	3.5	Coding	Django view dor deliveries
2015-03-31	Sigurbjörn	3	Coding	Retrieve build from parabuild
2015-03-31	Árni	3.5	Coding	
2015-04-04	Freyr	2	Testing	Unit tests for copy_delivery_files_to_storage_server script
2015-04-04	Gunnar	5	Coding	Djaongo list of delivery
2015-04-04	Sigurbjörn	4	Coding	Implementation of copy_delivery_files_to_storage_server script
2015-04-04	Árni	5	Coding	

Date	Who	Hours (1381)	Type	Description of what was done
2015-04-04	Freyr	5	Coding	Implementation of copy_delivery_files_to_storage_server script
2015-04-08	Gunnar	7	Meeting	Sprint 5 review, Retrospective and Sprint 6 planning
2015-04-08	Sigurbjörn	7	Meeting	Sprint 5 review, Retrospective and Sprint 6 planning
2015-04-08	Árni	7	Meeting	Sprint 5 review, Retrospective and Sprint 6 planning
2015-04-08	Freyr	5	Meeting	Sprint 5 review, Retrospective and Sprint 6 planning
2015-04-09	Freyr	2	Other	Investigate how to use Perforce API for task header scripts
2015-04-11	Freyr	8	Testing	Unit tests for task header date parser and task header parser
2015-04-11	Sigurbjörn	7	Coding	
2015-04-13	Freyr	6	Testing	Unit tests for task header date parser and task header parser
2015-04-13	Sigurbjörn	5	Other	Creating regex to find first line of each task header entry
2015-04-14	Gunnar	9	Coding	get_task_from_targetprocess scriptan authentication
2015-04-14	Sigurbjörn	8	Coding	Work on Task Header Parser, including regex work
2015-04-14	Árni	8	Coding	
2015-04-14	Freyr	8	Coding	Implementation of task header parser
2015-04-15	Árni	8	Other	
2015-04-15	Gunnar	9	Coding	get_task_from_targetprocess
2015-04-15	Sigurbjörn	5	Other	Regex work for task header parser
2015-04-15	Sigurbjörn	4	Coding	task header parser and get tasks from targetprocess
2015-04-15	Freyr	5	Testing	Unit tests for task header parser
2015-04-15	Freyr	4	Coding	Implementation of task header parser
2015-04-16	Sigurbjörn	6	Coding	implementation of get_tasks_from_targetprocess
2015-04-18	Árni	8	Other	html, css for delivery view
2015-04-18	Freyr	10	Coding	Work on check_task_header_incremental_nr script
2015-04-18	Gunnar	10	Other	html, css for delivery view
2015-04-18	Sigurbjörn	8	Coding	Work on check_task_header_incremental_nr script
2015-04-20	Árni	6	Meeting	Sprint 6 review, Retrospective and Sprint 7 planning
2015-04-20	Gunnar	6	Meeting	Sprint 6 review, Retrospective and Sprint 7 planning
2015-04-20	Freyr	6	Meeting	Sprint 6 review, Retrospective and Sprint 7 planning
2015-04-20	Sigurbjörn	6	Meeting	Sprint 6 review, Retrospective and Sprint 7 planning
2015-04-26	Árni	4	Documentation	Updating Project Schedule and Development Report
2015-04-27	Sigurbjörn	4	Coding	Starting work on check_task_header_valid_tasks script
2015-04-27	Árni	8	Documentation	Update reports for status meeting 3 handin
2015-04-27	Gunnar	9	Coding	copy_delivery_to_storage_server
2015-04-27	Freyr	9	Coding	Updating create_delivery script (previously delivery_automation_tool.py)
2015-04-28	Sigurbjörn	6	Coding	Starting work on check_task_header_valid_tasks script
2015-04-28	Árni	8	Other	
2015-04-28	Gunnar	9	Coding	copy_delivery_to_storage_server
2015-04-28	Freyr	8	Coding	Updating create_delivery script (previously delivery_automation_tool.py)
2015-04-29	Árni	9	Coding	retrieve_documentation_from_sharepoint
2015-04-29	Sigurbjörn	9	Testing	Working on unit tests for check_tash_header_valid_tasks
2015-04-29	Gunnar	9	Coding	retrieve_documentation_from_sharepoint
2015-04-29	Freyr	3	Documentation	Updating operational manual and user guide
2015-04-29	Freyr	4	Testing	Updating unit tests to use actual mock objects
2015-04-29	Freyr	3	Coding	Various fixes to scripts
2015-04-30	Árni	10	Coding	retrieve_documentation_from_sharepoint
2015-04-30	Gunnar	10	Coding	working on retrive_documents_from_sharepoint
2015-04-30	Sigurbjörn	4	Other	Investigate ways to mock perforce connection in unit tests + help with sharepoint script
2015-04-30	Freyr	8	Testing	Updating unit tests to use actual mock objects
2015-04-30	Sigurbjörn	6	Coding	check_task_header_valid_tasks
2015-05-01	Gunnar	10	Coding	working on retrive_documents_from_sharepoint
2015-05-01	Sigurbjörn	8	Other	
2015-05-01	Freyr	10	Testing	Unit tests for check_task_header_dates script
2015-05-03	Gunnar	10	Coding	
2015-05-03	Árni	7	Coding	
2015-05-03	Sigurbjörn	9	Testing	Finishing unit tests for valid_tasks script
2015-05-03	Freyr	10	Testing	Unit tests for check_task_header_dates script
2015-05-03	Árni	3	Documentation	
2015-05-03	Sigurbjörn	1	Coding	Minor adjustments to valid_tasks script
2015-05-04	Gunnar	8	Other	Script overview
2015-05-04	Sigurbjörn	2	Meeting	Sprint 7 review, Retrospective and Sprint 8 planning
2015-05-04	Freyr	2	Meeting	Sprint 7 review, Retrospective and Sprint 8 planning

Date	Who	Hours (1381)	Type	Description of what was done
2015-05-04	Árni	2	Meeting	Sprint 7 review, Retrospective and Sprint 8 planning
2015-05-04	Árni	8	Documentation	Updating files after latest sprint
2015-05-04	Gunnar	2	Meeting	Sprint 7 review, Retrospective and Sprint 8 planning
2015-05-04	Sigurbjörn	2	Coding	
2015-05-04	Sigurbjörn	6	Testing	Fixing bugs in unit tests for valid_tasks script
2015-05-04	Freyr	4	Coding	Update scripts using run vs. Script.execute
2015-05-05	Árni	9	Coding	Add view for delivery report
2015-05-05	Sigurbjörn	4	Other	Investigate how to get a searchable pdf from html view in django since pdfkit isn't sufficie
2015-05-05	Sigurbjörn	5	Coding	working on export_delivery_report_to_pdf script
2015-05-05	Gunnar	8	Coding	working on views for index and report
2015-05-05	Freyr	8	Coding	Update scripts using run vs. Script.execute
2015-05-06	Árni	5	Coding	Add view for delivery report
2015-05-06	Árni	6	Other	prepare presentation for Status Meeting 3
2015-05-06	Gunnar	5	Coding	working on views for index and report
2015-05-06	Gunnar	6	Other	prepare presentation for Status Meeting 3
2015-05-06	Freyr	8	Coding	Update scripts using run vs. Script.execute and logging and ISO script
2015-05-06	Sigurbjörn	7	Coding	working on export_delivery_report_to_pdf script
2015-05-06	Freyr	1	Meeting	Meeting with instructor
2015-05-06	Árni	1	Meeting	Meeting with instructor
2015-05-06	Gunnar	1	Meeting	Meeting with instructor
2015-05-06	Sigurbjörn	1	Meeting	Meeting with instructor
2015-05-06	Freyr	6	Testing	Updating unit tests after refactoring
2015-05-06	Sigurbjörn	2	Other	Investigating xhtml2pdf and reportlab documentation
2015-05-07	Gunnar	3	Meeting	Third Status Meeting HR + Retrospective
2015-05-07	Árni	3	Meeting	Third Status Meeting HR + Retrospective
2015-05-07	Sigurbjörn	3	Meeting	Third Status Meeting HR + Retrospective
2015-05-07	Freyr	3	Meeting	Third Status Meeting HR + Retrospective
2015-05-07	Gunnar	6	Coding	
2015-05-07	Sigurbjörn	5	Coding	
2015-05-07	Freyr	7	Testing	Unit tests for create_new_mod_del_files
2015-05-07	Árni	9	Coding	Retrieve_documentation_from_sharepoint
2015-05-08	Freyr	12	Coding	Unit tests for create_new_mod_del_files
2015-05-08	Sigurbjörn	5	Coding	Working on export_delivery_to_pdf
2015-05-08	Sigurbjörn	5	Other	
2015-05-08	Árni	9.5	Coding	Retrieve_documentation_from_sharepoint
2015-05-08	Gunnar	11	Coding	Retrive_documentation_from_sharepoint
2015-05-09	Árni	4	Coding	Retrieve_documentation_from_sharepoint
2015-05-10	Árni	6	Coding	Retrieve_documentation_from_sharepoint
2015-05-10	Sigurbjörn	9.5	Coding	Work on export_delivery_to_pdf and and report_pdf.html
2015-05-10	Freyr	10	Coding	Various fixes and updates to code
2015-05-10	Freyr	3	Testing	Updates of various unit tests
2015-05-10	Gunnar	4	Other	Reasearching deployment to production server
2015-05-10	Gunnar	7	Other	Reasearching deployment to production server
2015-05-11	Sigurbjörn	7	Coding	Refactoring urls.py, views.py and files under template folder
2015-05-11	Sigurbjörn	2	Testing	Writing unit tests for export_delivery_to_pdf and get_tasks_from_targetprocess
2015-05-11	Sigurbjörn	2	Documentation	Work on final report
2015-05-11	Freyr	10	Testing	Adding various missing unit tests
2015-05-11	Árni	10	Documentation	Update reports + work on final report
2015-05-11	Freyr	2	Documentation	Work on final report
2015-05-11	Sigurbjörn	1	Meeting	Sprint 8 review, Retrospective and Sprint 9 planning
2015-05-11	Árni	1	Meeting	Sprint 8 review, Retrospective and Sprint 9 planning
2015-05-11	Gunnar	1	Meeting	Sprint 8 review, Retrospective and Sprint 9 planning
2015-05-11	Freyr	1	Meeting	Sprint 8 review, Retrospective and Sprint 9 planning
2015-05-11	Gunnar	12	Other	Work on final report dicumentation, deploying
2015-05-11	Sigurbjörn	2	Other	Refactoring get_tasks, export_pdf and valid_tasks and messages.py
2015-05-12	Árni	12.5	Documentation	Final report
2015-05-12	Freyr	10	Testing	Release test of DATI
2015-05-12	Gunnar	10.5	Documentation	Final report
2015-05-12	Sigurbjörn	13	Documentation	Final report
2015-05-12	Freyr	4.5	Documentation	Final report
2015-05-13	Freyr	7	Documentation	Final report

Date	Who	Hours (1381)	Type	Description of what was done
2015-05-13	Árni	8	Documentation	Final report
2015-05-13	Gunnar	7	Documentation	Final report
2015-05-13	Sigurbjörn	8	Documentation	Final report

Overall

	Coding	Documentation	Meeting	Other	Testing	Total
Árni	121.5	96	56.5	40	13.5	327.5
Freyr	117.5	77	55	12	117	378.5
Gunnar	153.5	53	54.5	52	17	330
Sigurbjörn	124.5	74.5	57	37	48	341
Total	517	300.5	223	141	195.5	1377

Sprint 0

	Coding	Documentation	Meeting	Other	Testing	Total
Árni	0	8.5	5.5	2	0	16
Freyr	0	8.5	5.5	4.5	0	18.5
Gunnar	0	8.5	5.5	0	0	14
Sigurbjörn	0	8.5	5.5	0	0	14
Total	0	34	22	6.5	0	62.5

Sprint 1 - Grimlock

	Coding	Documentation	Meeting	Other	Testing	Total
Árni	8	7.5	7.5	0	0	23
Freyr	6.5	7.5	7.5	0	2	23.5
Gunnar	6.5	7.5	7.5	0	2	23.5
Sigurbjörn	7	7.5	7.5	0	0	22
Total	28	30	30	0	4	92

Sprint 2 - Bumblebee

	Coding	Documentation	Meeting	Other	Testing	Total
Árni	6	8.5	11.5	0	0	26
Freyr	5	8.5	12	0	0	25.5
Gunnar	5	8.5	11.5	0	0	25
Sigurbjörn	6	6	12	0	0	24
Total	22	31.5	47	0	0	100.5

Sprint 3 - Megatron

	Coding	Documentation	Meeting	Other	Testing	Total
Árni	5	0	4	3	11.5	23.5
Freyr	4.5	6.5	4	0	14.5	29.5
Gunnar	0	6.5	4	0	15	25.5
Sigurbjörn	10	2.5	4	0	12.5	29
Total	19.5	15.5	16	3	53.5	107.5

Sprint 4 - Ironhide

	Coding	Documentation	Meeting	Other	Testing	Total
Árni	19	15.5	5	0	2	41.5
Freyr	3	29.5	5	5.5	4	47
Gunnar	17	4.5	3	5	0	29.5
Sigurbjörn	0	27	5	2	2	36
Total	39	76.5	18	12.5	8	154

Sprint 5 - Jazz

	Coding	Documentation	Meeting	Other	Testing	Total
Árni	16	2.5	3	0	0	21.5
Freyr	14.5	0	3	0	9.5	27
Gunnar	20	0	3	0	0	23
Sigurbjörn	11	0	3	0	7.5	21.5
Total	61.5	2.5	12	0	17	93

Sprint 6 - Optimus Prime

	Coding	Documentation	Meeting	Other	Testing	Total
Árni	8	0	7	16	0	31
Freyr	22	0	5	2	19	48
Gunnar	18	0	7	10	0	35
Sigurbjörn	33	0	7	10	0	50
Total	81	0	26	38	19	164

Sprint 7 - Hot Rod

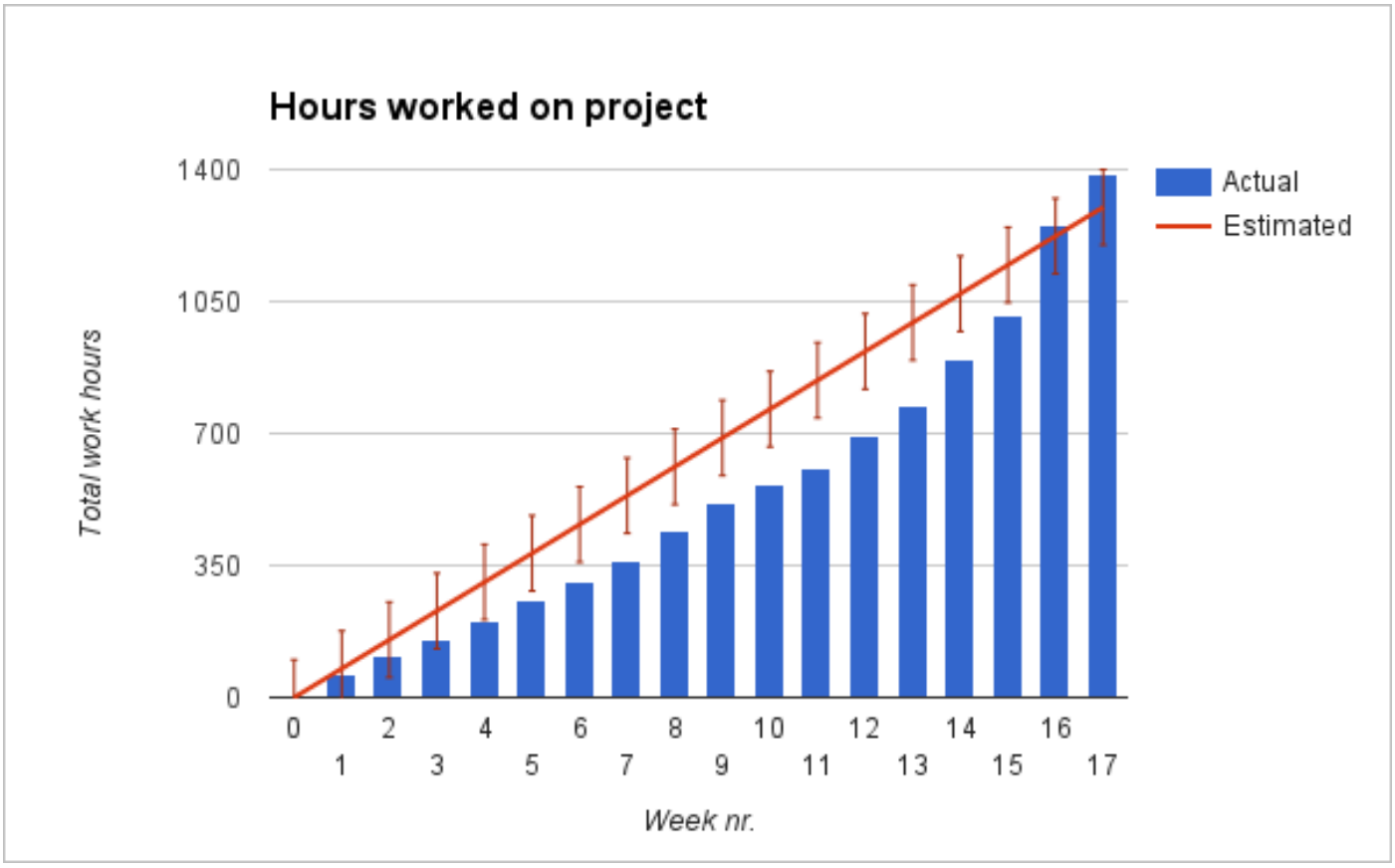
	Coding	Documentation	Meeting	Other	Testing	Total
Árni	26	15	6	8	0	55
Freyr	20	3	6	0	32	61
Gunnar	57	0	6	0	0	63
Sigurbjörn	25	0	6	12	18	61
Total	128	18	24	20	50	240

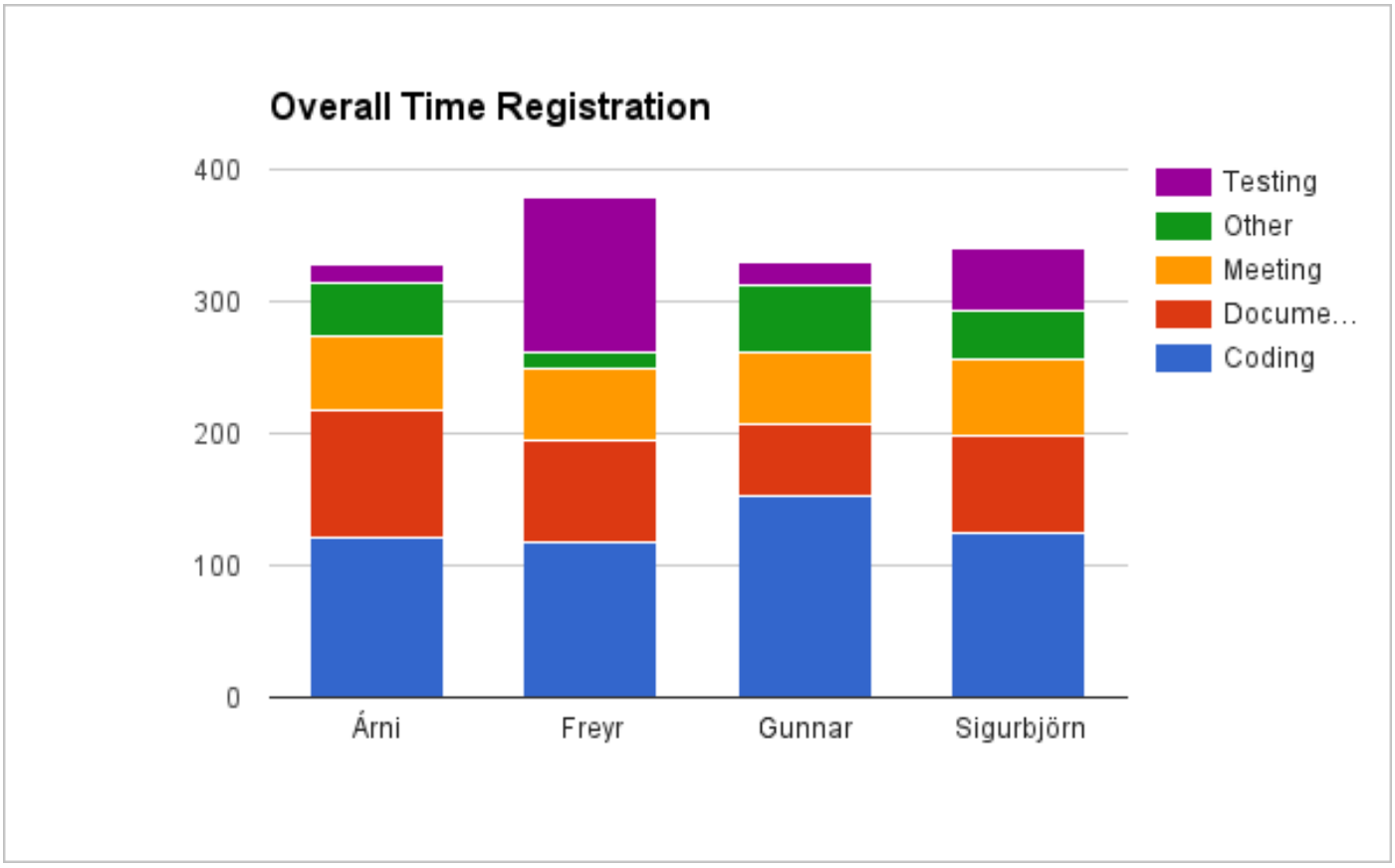
Sprint 8 - Mirage

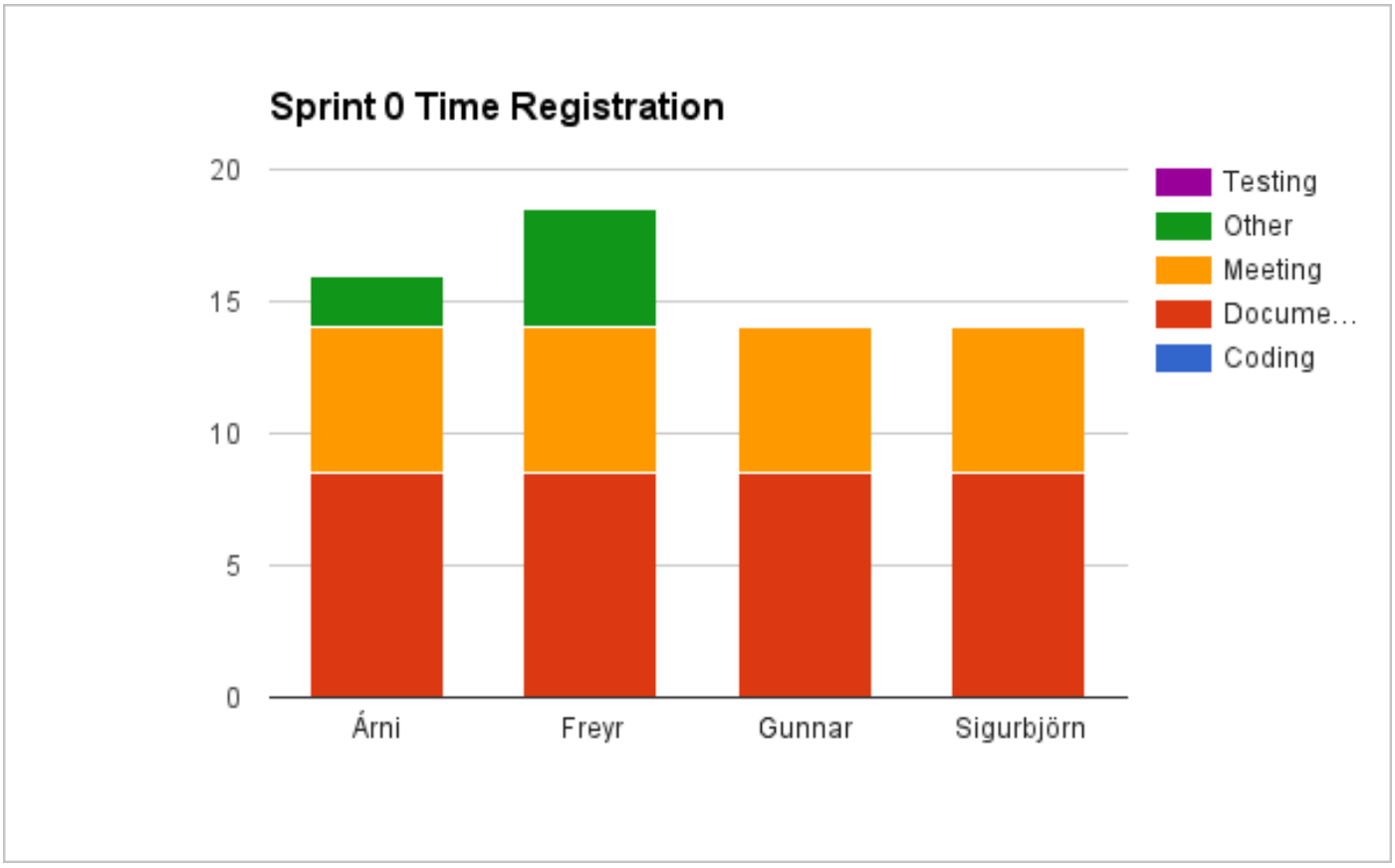
	Coding	Documentation	Meeting	Other	Testing	Total
Árni	33.5	8	6	11	0	58.5
Freyr	42	0	6	0	16	64
Gunnar	30	0	6	25	0	61
Sigurbjörn	33.5	0	6	11	6	56.5
Total	139	8	24	47	22	240

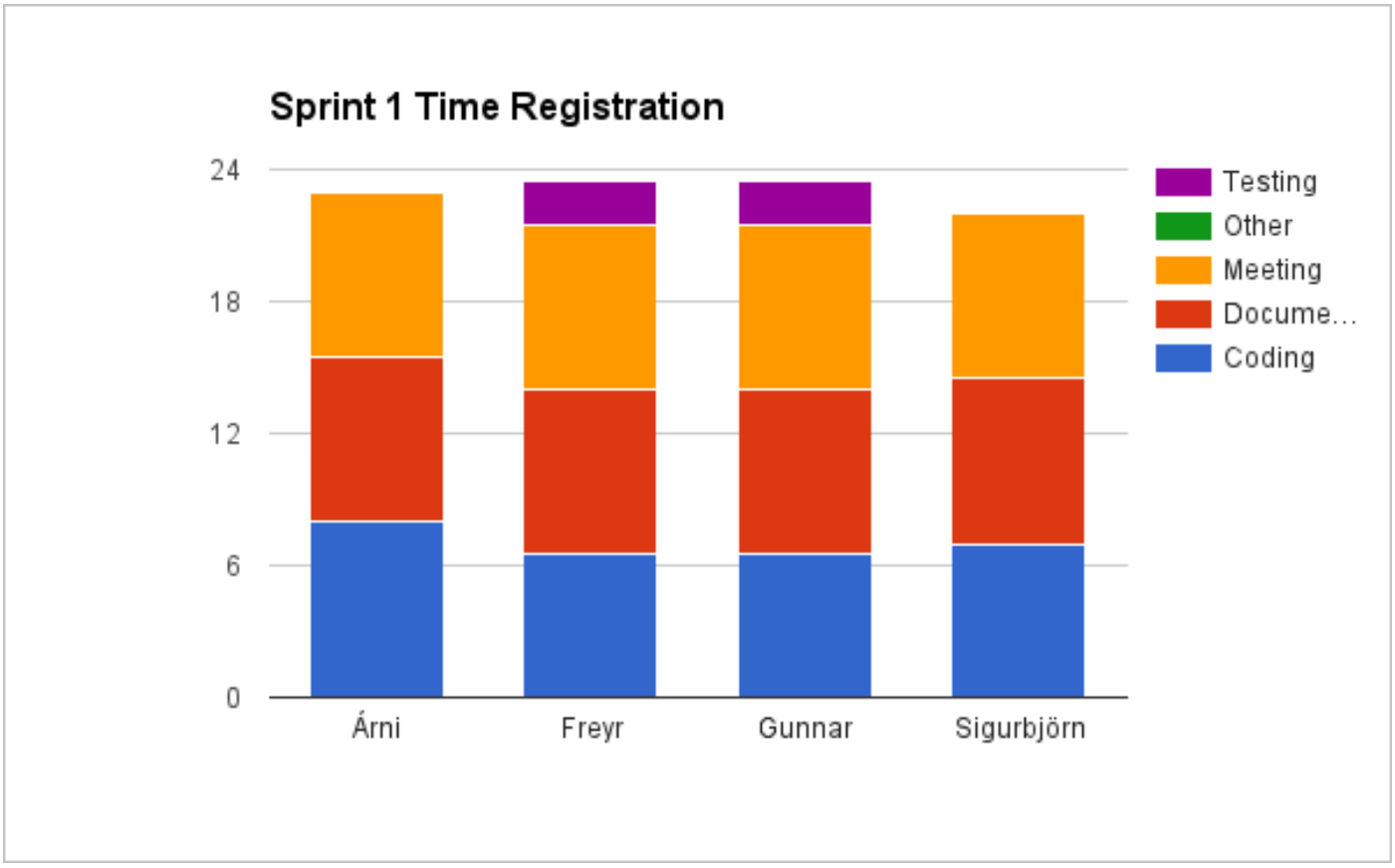
Sprint 9 - Ultra Magnus

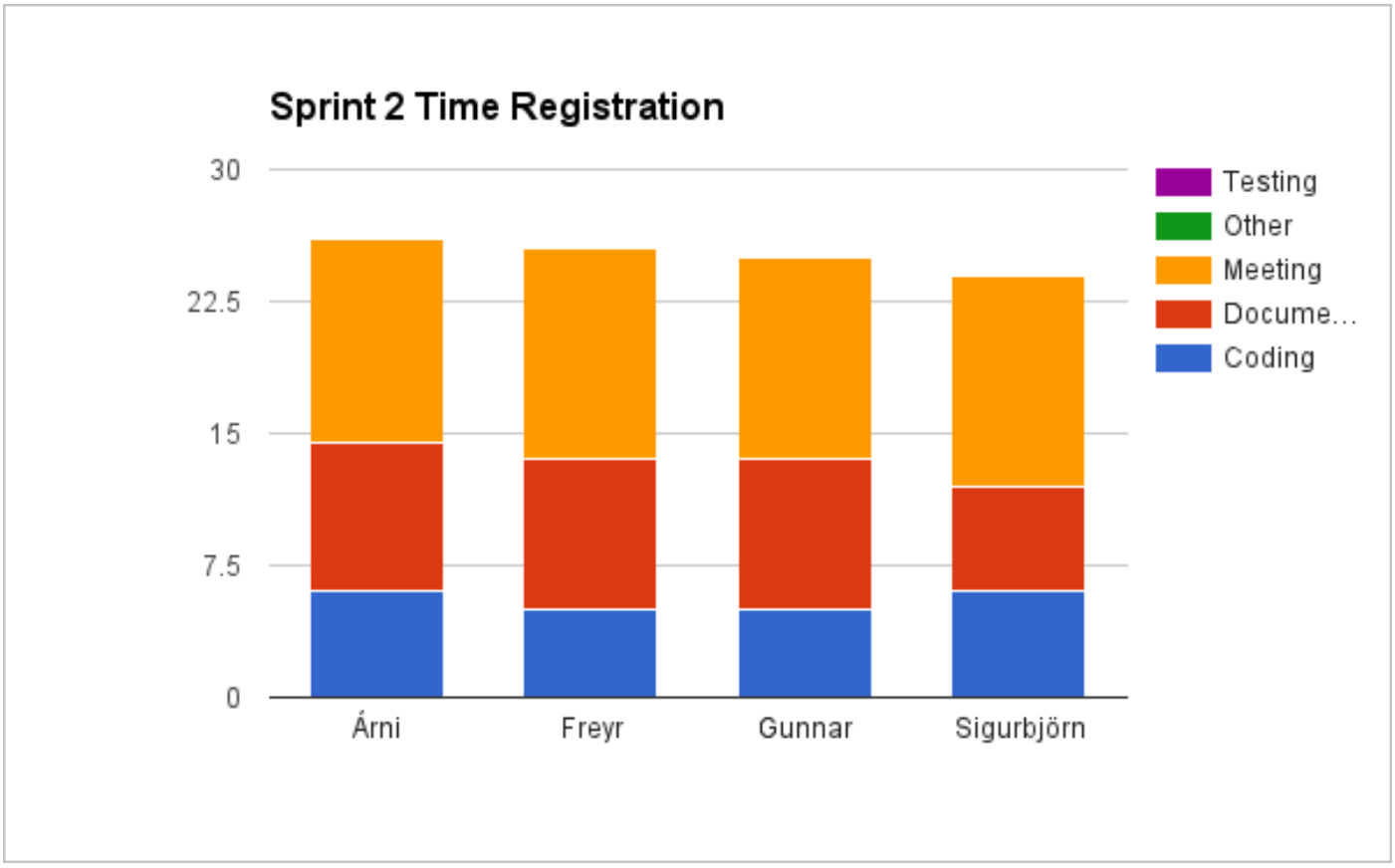
	Coding	Documentation	Meeting	Other	Testing	Total
Árni	0	30.5	1	0	0	31.5
Freyr	0	13.5	1	0	20	34.5
Gunnar	0	17.5	1	12	0	30.5
Sigurbjörn	7	23	1	2	2	35
Total	7	84.5	4	14	22	131.5

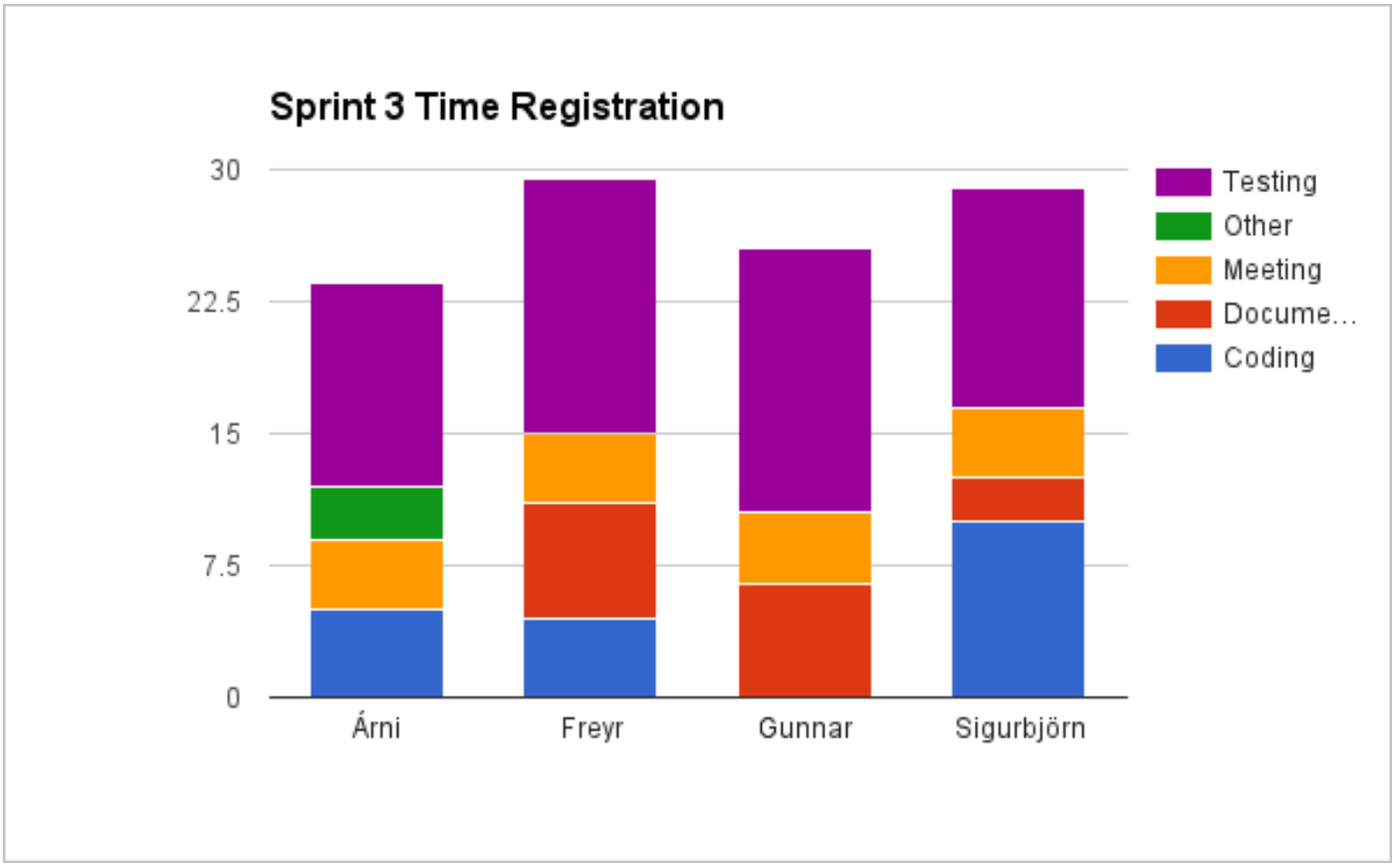

Hours worked on project
1400
1050
700
350
0
Total work hours
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
Week nr.
Actual
Estimated

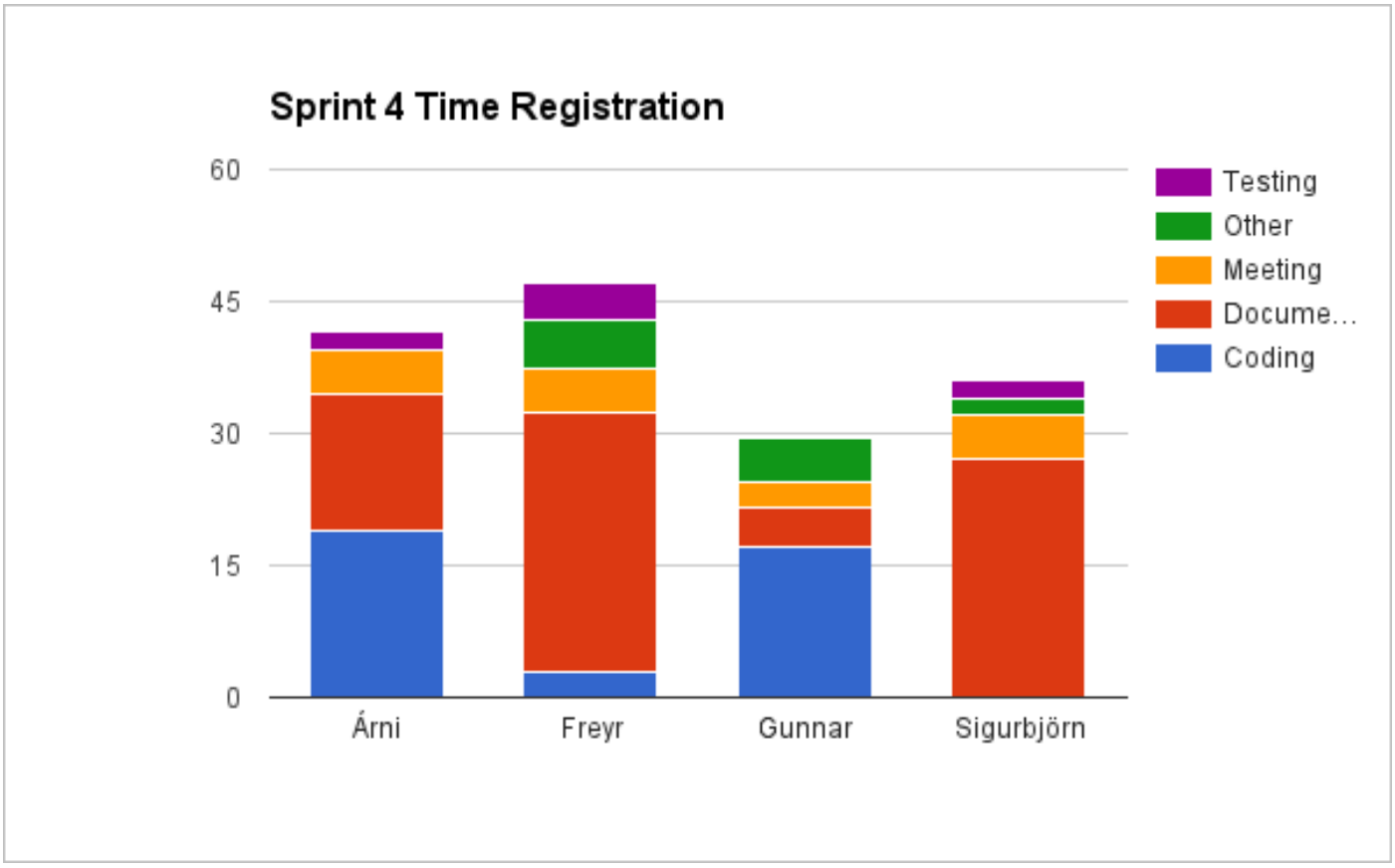

Overall Time Registration
400
300
200
100
0
Árni
Freyr
Gunnar
Sigurbjörn
Testing
Other
Meeting
Docume...
Coding

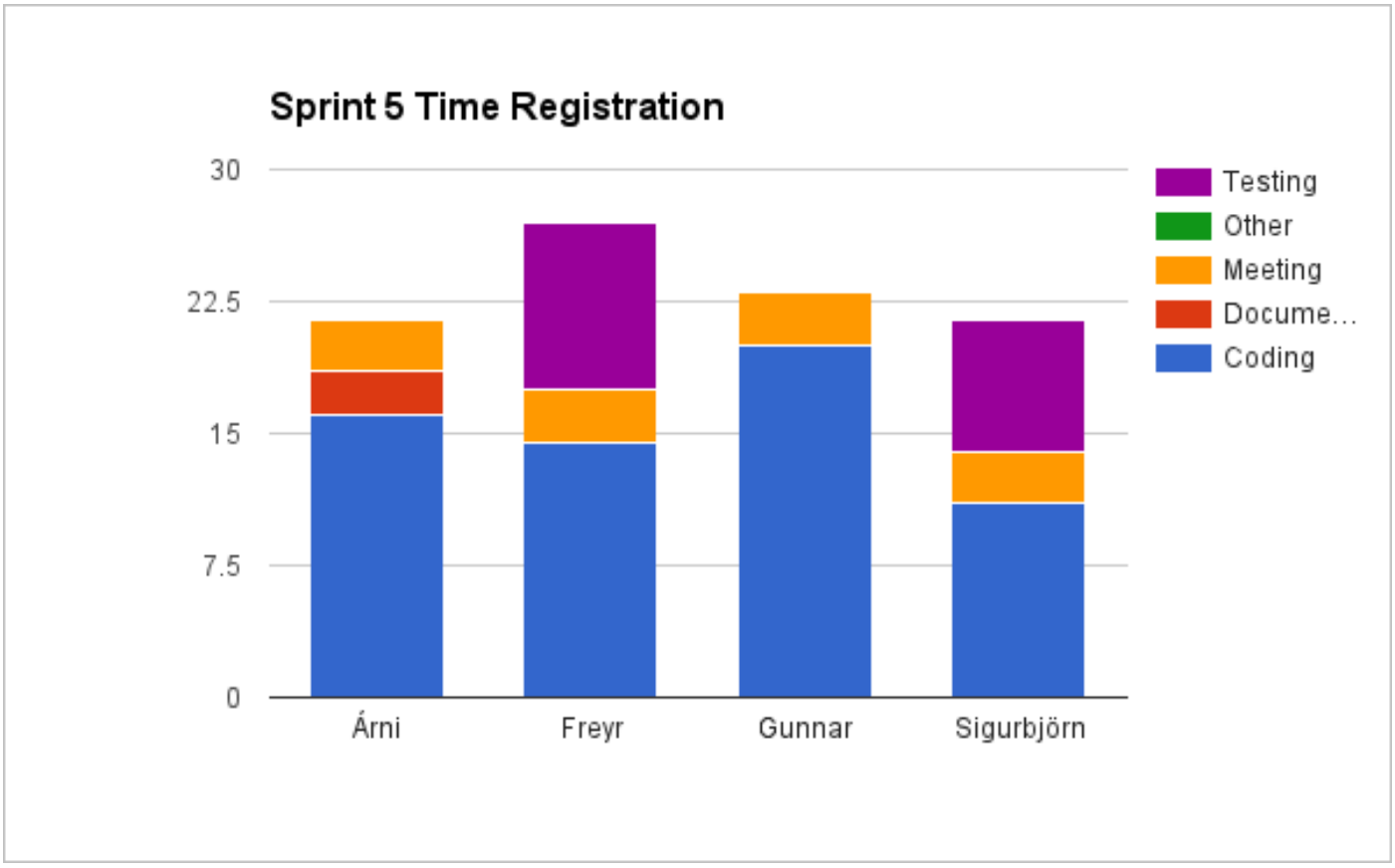

Sprint 0 Time Registration
20
15
10
5
0
Árni
Freyr
Gunnar
Sigurbjörn
Testing
Other
Meeting
Docume...
Coding

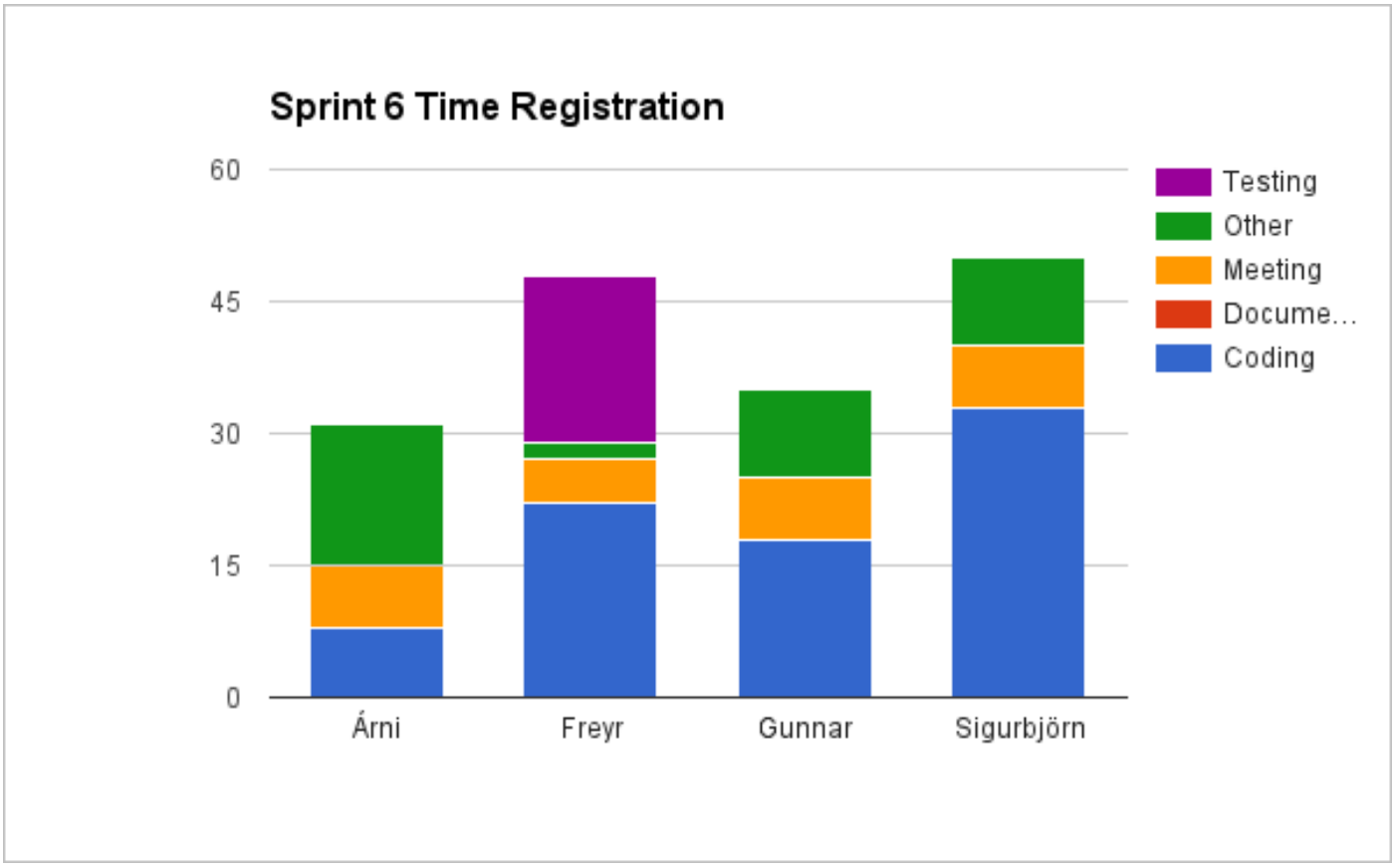

Sprint 1 Time Registration
24
18
12
6
0
Árni
Freyr
Gunnar
Sigurbjörn
Testing
Other
Meeting
Docume...
Coding

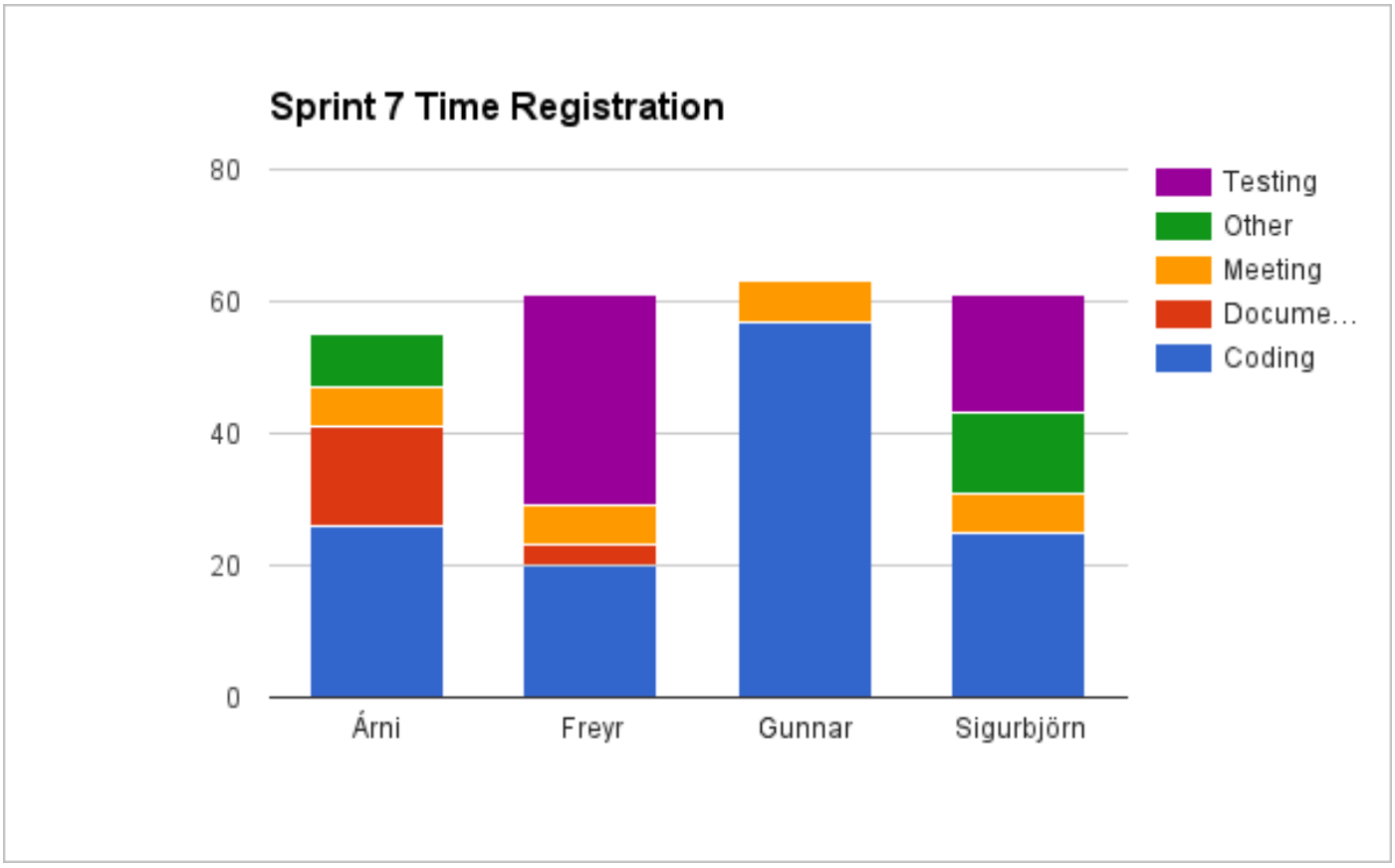

Sprint 2 Time Registration
30
22.5
15
7.5
0
Árni
Freyr
Gunnar
Sigurbjörn
Testing
Other
Meeting
Docume...
Coding

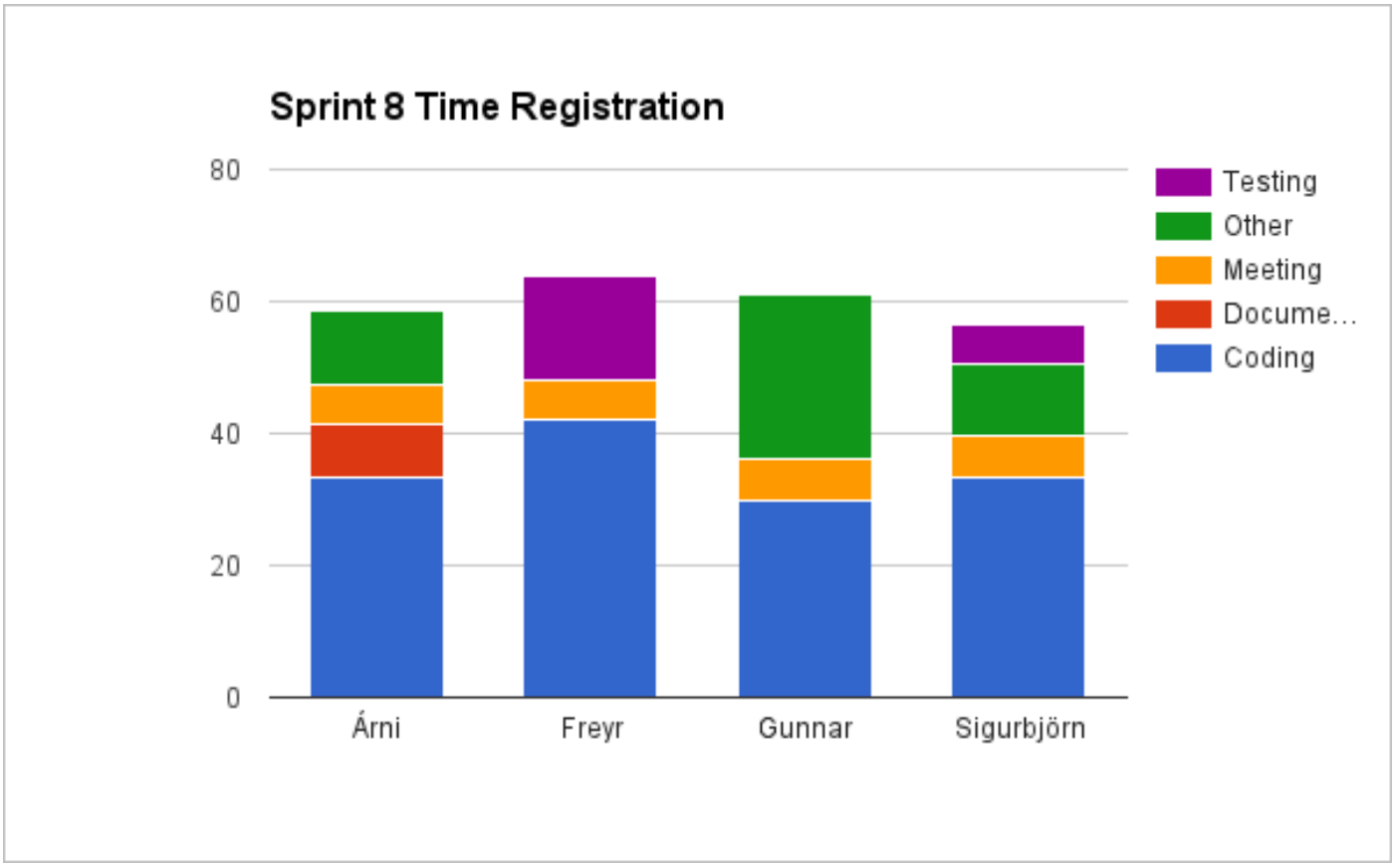

Sprint 3 Time Registration
30
22.5
15
7.5
0
Árni
Freyr
Gunnar
Sigurbjörn
Testing
Other
Meeting
Docume...
Coding

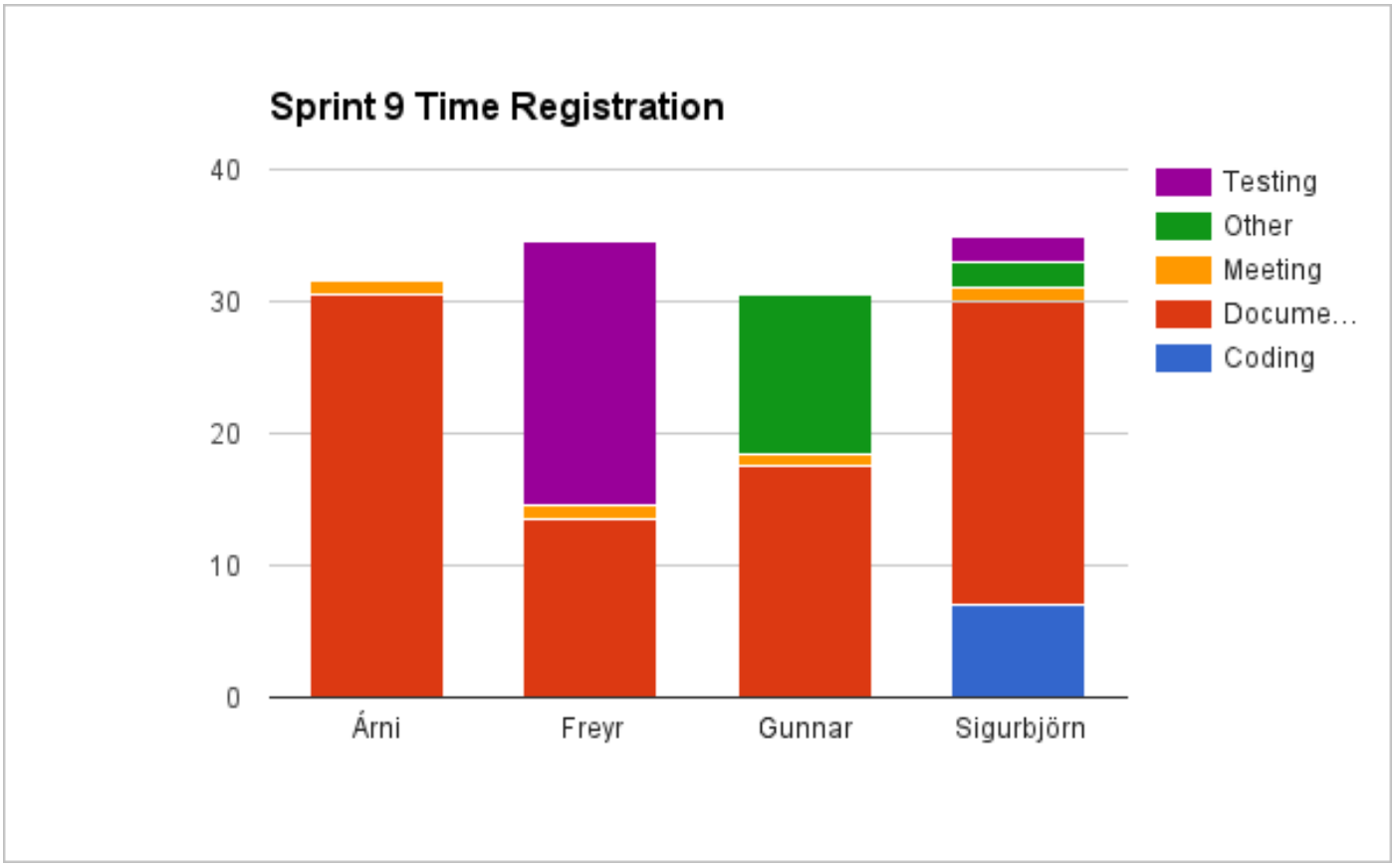

Sprint 4 Time Registration
60
45
30
15
0
Árni
Freyr
Gunnar
Sigurbjörn
Testing
Other
Meeting
Docume...
Coding


Sprint 5 Time Registration
30
22.5
15
7.5
0
Árni
Freyr
Gunnar
Sigurbjörn
Testing
Other
Meeting
Docume...
Coding


Sprint 6 Time Registration
60
45
30
15
0
Árni
Freyr
Gunnar
Sigurbjörn
Testing
Other
Meeting
Docume...
Coding


Sprint 7 Time Registration
80
60
40
20
0
Árni
Freyr
Gunnar
Sigurbjörn
Testing
Other
Meeting
Docume...
Coding


Sprint 8 Time Registration
80
60
40
20
0
Árni
Freyr
Gunnar
Sigurbjörn
Testing
Other
Meeting
Docume...
Coding


Sprint 9 Time Registration
40
30
20
10
0
Árni
Freyr
Gunnar
Sigurbjörn
Testing
Other
Meeting
Docume...
Coding

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

Tern Systems - Project Schedule

T-404-LOKA-2015

Students:
Árni Þorvaldsson (arnitho12@ru.is)
Freyr Bergsteinsson (freyrb12@ru.is)
Gunnar Þór Helgason (gunnarh07@ru.is)
Sigurbjörn Kristjánsson (sigurbjorn12@ru.is)

Instructor:
Daníel Máni Jónsson (daniel@valitor.is)

Examiner:
Hannes Pétursson (hap@ru.is)

Table of Contents

Document History

Date	Author	Description
2015-01-27	ÁÞ, FB, GÞH, SK	Initial version. Includes product backlog items and a description of what's done in sprint 1.
2015-02-05	ÁÞ, FB, GÞH, SK	Updated backlog and added information about how the project is being developed.
2015-02-09	ÁÞ, FB, GÞH, SK	Updated backlog, added sprint 2 and added an overview of what was done in sprint 1.
2015-03-12	ÁÞ	Updated backlog, added tasks and burndown chart to sprint 2, added sprints 3 and 4
2015-03-29	ÁÞ	Updated backlog, updated sprint 4 and added user stories to sprint 5
2015-04-08	ÁÞ	Updated backlog, updated sprint 5 and add user stories to sprint 6
2015-04-26	ÁÞ	Updated backlog, updated sprint 6 and added user stories to sprint 7.
2015-05-04	ÁÞ	Updated backlog, updated sprint 7
2015-05-13	ÁÞ	Added sprints 8 and 9

Development

The team utilizes TDD (test driven development) while working on the project. Daily stand-up meeting are held virtually via Google Hangout at 13:55 on the days when the team is not together, otherwise the daily stand-up is held when each work session starts.

Product backlog

Story points estimation

The user stories' story points were estimated with Planning Poker before Sprint #1 began. During all subsequent sprint plannings, the backlog is reviewed and updated according to our experience with story point estimation and the velocity of the team (backlog grooming).

Backlog maintenance

The product backlog is maintained in Visual Studio Online. User stories that are selected for a sprint are removed from the backlog and assigned accordingly.

User stories in the product backlog

(updated May 4th)

ID	Title	Work Item Type	State	Effort	Iteration Path	Tags
121	As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	Product Backlog Item	Committed	1	Delivery Automation Tool for Isavia\Release 1\Sprint 4 - Ironhide	

20	As a user I want to make sure that the task headers delivered contain valid task numbers so that the delivery isn't rejected	Product Backlog Item	Committed	3	Delivery Automation Tool for Isavia\Release 1\Sprint 7 - Hot Rod
18	As a user I want task headers to show the correct date in task headers so that the delivery isn't rejected	Product Backlog Item	Committed	5	Delivery Automation Tool for Isavia\Release 1\Sprint 7 - Hot Rod
214	As a user I want to retrieve all documents from a folder in SharePoint for a product instead of one document	Product Backlog Item	Committed	3	Delivery Automation Tool for Isavia\Release 1\Sprint 7 - Hot Rod
213	As a user I don't want old backup files to be overwritten in case of mistakes in new delivery	Product Backlog Item	Committed	3	Delivery Automation Tool for Isavia\Release 1\Sprint 7 - Hot Rod
70	As an instructor and examiner I want to see a presentation of the product for the third status meeting (Stöðufundur 3) so I can assess the progress of the project	Product Backlog Item	Committed	1	Delivery Automation Tool for Isavia\Release 1\Sprint 7 - Hot Rod

217	As a user I want to be able to create a new delivery with a command line script or perform re-delivery on an existing delivery	Product Backlog Item	Committed	3	Delivery Automation Tool for Isavia\Release 1\Sprint 7 - Hot Rod
155	As a system administrator I want my project in a virtual environment so that it's easier to deploy it	Product Backlog Item	Approved	2	Delivery Automation Tool for Isavia\Release 1\Sprint 8 - Mirage
19	As a user I want to generate a delivery report in PDF format so that I can add it to the delivery	Product Backlog Item	Approved	3	Delivery Automation Tool for Isavia\Release 1\Sprint 8 - Mirage
76	As an instructor and examiner I want to see the project documents updated with regards to points made in Status Meeting 3	Product Backlog Item	Approved	1	Delivery Automation Tool for Isavia\Release 1\Sprint 8 - Mirage
71	As an instructor and examiner I want to see a final report for the project so I can grade the project	Product Backlog Item	Approved	8	Delivery Automation Tool for Isavia\Release 1\Sprint 9 - Ultra Magnus

159	As a web user I want to see the progress of the delivery process so that I can get a better visual clue	Product Backlog Item	New	8	Delivery Automation Tool for Isavia
22	As a user I want to see tasks in a release as defined in JIRA so that I can verify that they are included in the delivery	Product Backlog Item	New	8	Delivery Automation Tool for Isavia

Sprint 1 (Grimlock), January 26th - February 8th

Planning

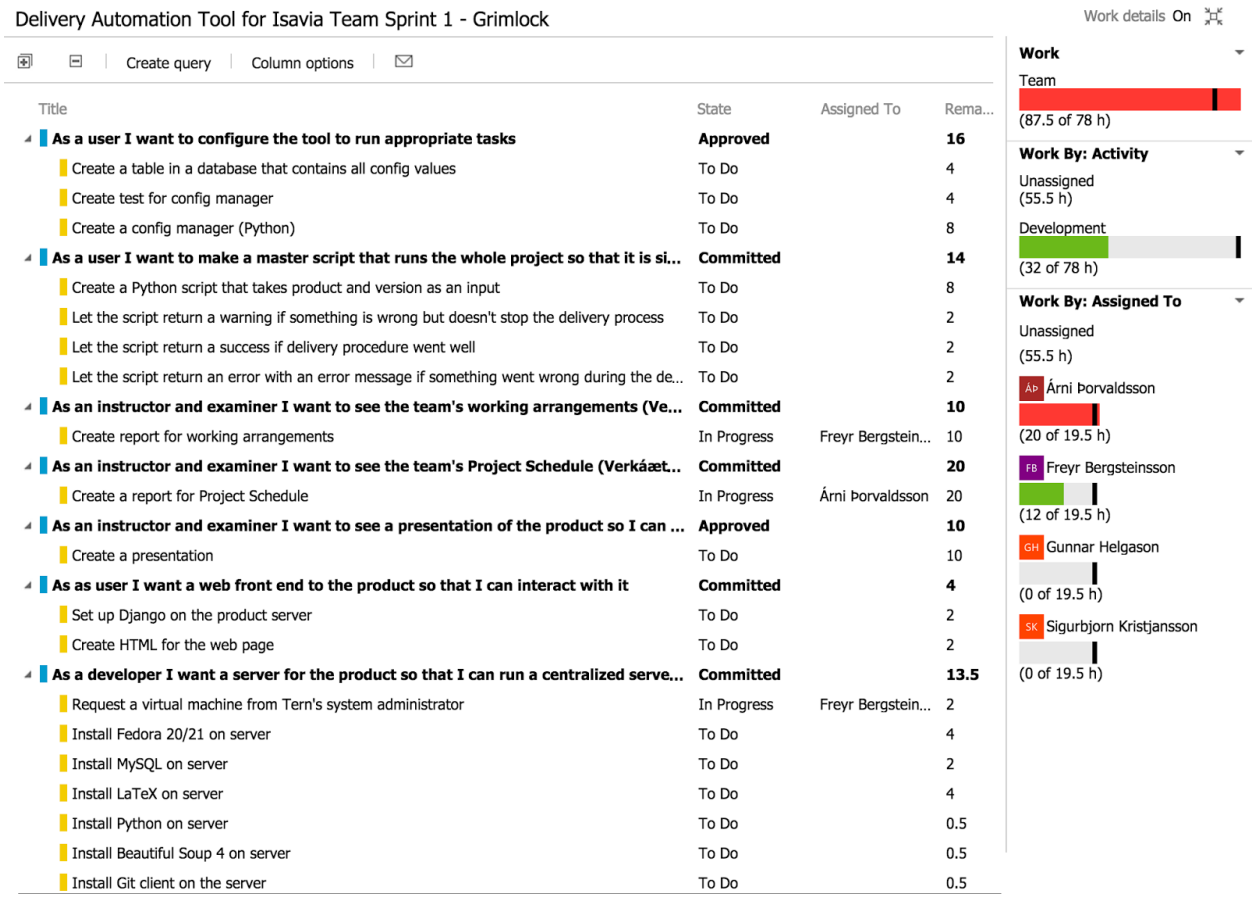
When planning the sprint we prioritized the user stories according to what's called MoSCoW ("Must have, Should have, Could have, Won't have"). Then we broke down each user story into as small tasks as we could and estimated the remaining work (in hours) for each task. We did not add any user stories to any later sprints at this time because we first want to get a feel for how well our estimations hold up and whether the velocity for this sprint is a good forecast for future sprints.

User stories

User story	Estimated story points
As a user I want to configure the tool to run appropriate tasks	8
As a user I want to make a master script that runs the whole project so that it is simpler to create delivery	2
As an instructor and examiner I want to see the team's working arrangements (Verkskipulag) so I can assess the progress of the project	1
As an instructor and examiner I want to see the team's Project Schedule (Verkáætlun) so I can assess the progress of the project	1
As an instructor and examiner I want to see a presentation of the product so I can assess the progress of the project	1
As as user I want a web front end to the product so that I can interact with it	2
As a developer I want a server for the product so that I can run a centralized server for web front end and running of scripts	2

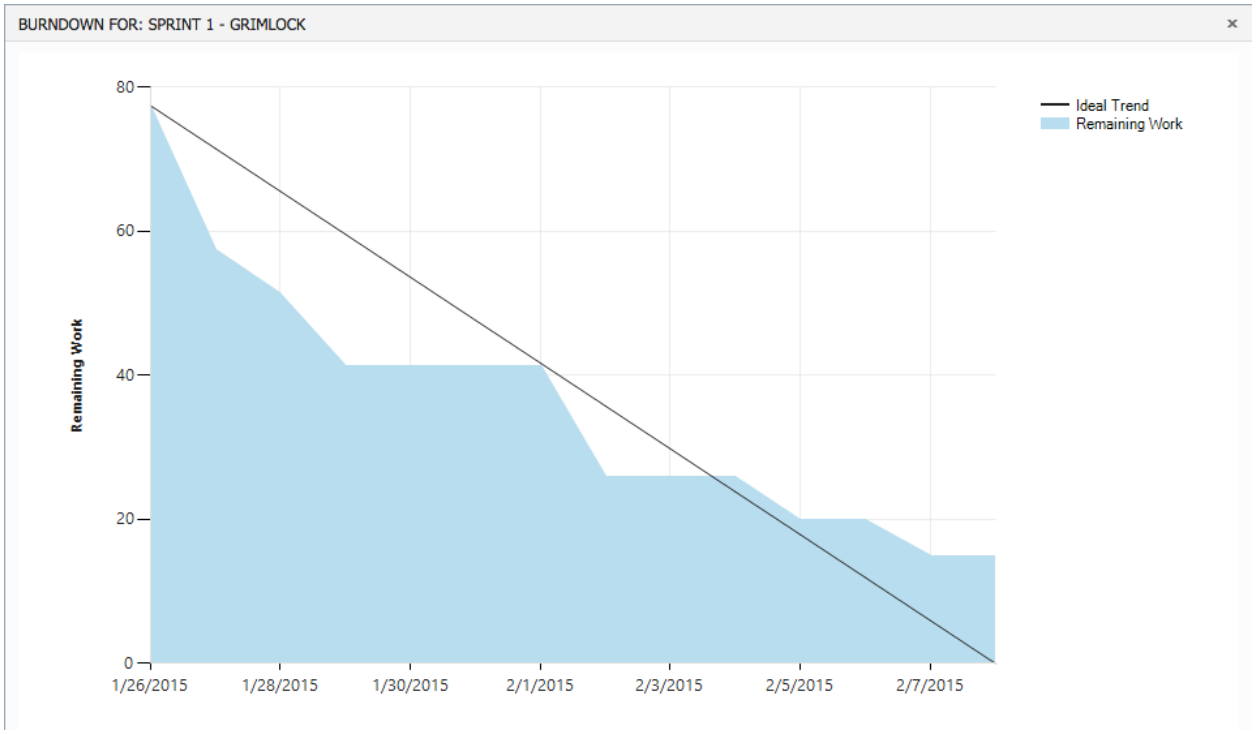
Tasks

Screenshot from Visual Studio Online showing task breakdown



Burndown Chart

Screenshot of the burndown chart after the sprint was finished:



The reason the chart does not reach 0 is due to one user story being removed after the sprint was over (Visual Studio Online does not reflect this update retroactively).

Sprint 2 (Bumblebee), February 9th - 22nd

Planning

One story in the previous sprint was left unfinished and is therefore scheduled to be done during this sprint. The MoSCoW method was again used to prioritize the user stories and the stories broken down into tasks as with the previous sprint.

User stories

User story	Estimated story points
As a user I want to make a master script that runs the whole project so that it is simpler to create delivery	5
As an instructor and examiner I want to see the team's Developments Report (Framvinduyfirlit) so I can assess the progress of the project	2
As an instructor and examiner I want to see the team's Risk Analysis (Áhættugreining) so I can assess the progress of the project	2
As an instructor and examiner I want to see the team's updated Project Schedule (Verkáætlun) so I can assess the progress of the project	1
As a developer I want a better overview of the internal working of the product so that I better understand it	3
As a scrum master I want to see scrum tools in order so that we can report work progress (Framvinduskýrsla)	1

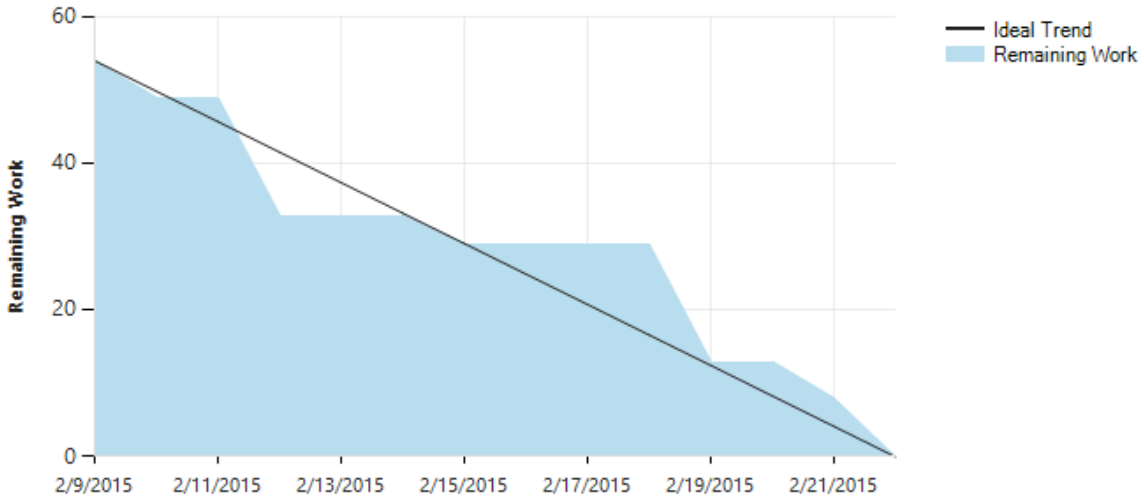
Tasks

Delivery Automation Tool for Isavia Team - Sprint 2 - Bumblebee - Backlog

ID	Title	State	Assigned To	Remaining Work
68	As an instructor and examiner I want to see a presentation of the product for the first status meeting (Stöðufundur 1) so I can assess the progress of the project	Done		
72	Update the presentation used the last time to include risk analysis and development report	Done	Freyr Bergsteinsson	
73	Rehearse the presentation	Done		
58	As an instructor and examiner I want to see the team's Developments Report (Framvinduyfirlit) so I can assess the progress of the project	Done		
60	Create a Development Report (Framvinduskýrsla)	Done	Árni Þorvaldsson	
63	Show how much time has been spent in total, per type and per member	Done	Freyr Bergsteinsson	
57	As an instructor and examiner I want to see the team's Risk Analysis (Áhættugreining) so I can assess the progress of the project	Done		
61	Create a Risk Analysis Report (Áhættugreining)	Done	Gunnar Helgason	
56	As an instructor and examiner I want to see the team's updated Project Schedule (Verkáætlun) so I can assess the progress of the project	Done		
62	Update the Project Schedule (Verkáætlun)	Done	Sigurbjorn Kristjansson	
55	As a developer I want a better overview of the internal working of the product so that I better understand it	Done		
64	Create a sequence diagram of the master script	Done	Freyr Bergsteinsson	

66	Create a schema of the database for the configuration manager	Done	Freyr Bergsteinsson
67	Create a schema of the delivery report database	Done	Freyr Bergsteinsson
29	As a user I want to make a master script that runs the whole project so that it is simpler to create delivery	Done	
39	Create a Python script that takes product and version as an input	Done	Árni Þorvaldsson
40	Let the script return a warning if something is wrong but doesn't stop the delivery process	Done	Sigurbjorn Kristjansson
41	Let the script return a success if delivery procedure went well	Done	Árni Þorvaldsson
42	Let the script return an error with an error message if something went wrong during the deliver procedure	Done	Sigurbjorn Kristjansson

Burndown chart



Sprint 3 (Megatron), February 23rd - March 8th

Planning

The MoSCoW method was used to prioritize the user stories in the backlog and user stories were then selected for the sprint with the velocity of the previous sprint in mind. The user stories were then broken into the following tasks and an estimation of working hours was made.

User stories

User story	Estimated story points
As a developer I want to easily add unit tests to the product so that the product is more reliable	2
As a developer I want the config manager in a separate database than the delivery report as to keep the tool separate from the data	2
As a developer I want the Config Manager to be part of the Django framework so that it is easier to maintain	4
As an instructor and examiner I want to see the project documents updated with regards to points made in Status Meeting 1	1
As a user I want delivery reports to be stored in a database so I can use it to generate a report later	2
As an instructor and examiner I want to see a draft of a design document so that I can better understand how the product is supposed to work	3

Tasks

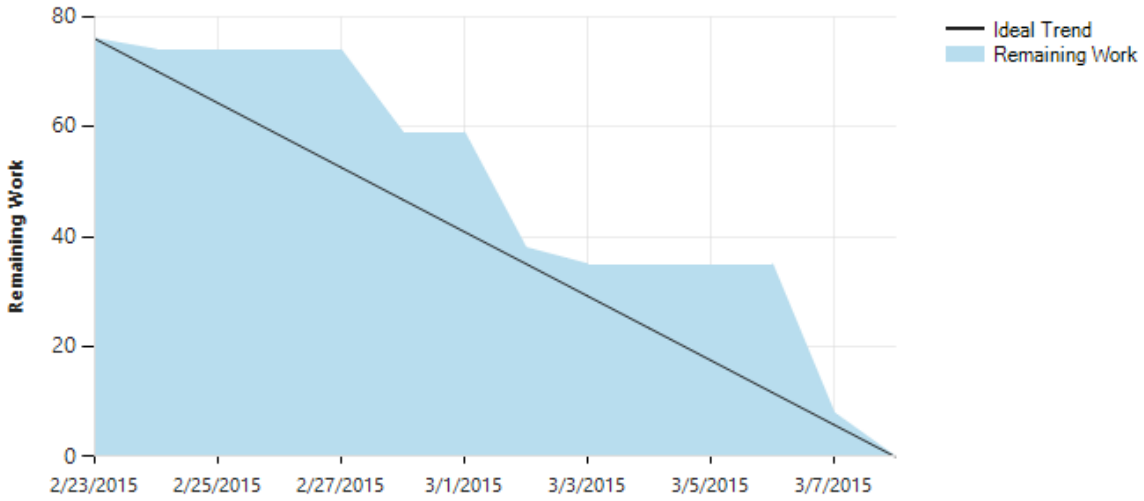
Delivery Automation Tool for Isavia Team - Sprint 3 - Megatron - Backlog

ID	Title	State	Assigned To	Remaining Work
35	As a developer I want to easily add unit tests to the product so that the product is more reliable	Done		
80	Investigate how unit tests are made with Django	Done	Freyr Bergsteinsson	
75	As a developer I want the config manager in a separate database than the delivery report as to keep the tool separate from the data	Done		
87	Investigate how to make Django use different databases	Done	Árni Þorvaldsson	
88	Split delivery report model from config manager model	Done	Freyr Bergsteinsson	
118	Split config manager into two apps, one for the config values and one for the product catalog	Done	Freyr Bergsteinsson	
59	As a developer I want the Config Manager to be part of the Django framework so that it is easier to maintain	Done		
104	Implement unit test for add function in the config manager	Done	Gunnar Helgason	
105	Implement unit test for the get function in the config manager	Done	Gunnar Helgason	
106	Implement unit test for the update function in the config manager	Done	Freyr Bergsteinsson	
107	Implement unit test for the delete function in the config manager	Done	Freyr Bergsteinsson	
82	Implement add function in the config manager that stores a new config value	Done	Gunnar Helgason	
83	Implement get function in the config manager that retrieves a value for some key	Done	Freyr Bergsteinsson	
84	Implement update function that updates the value for some key in the config manager	Done	Freyr Bergsteinsson	

85	Implement a delete function in the config manager that removes a key/value pair from the database	Done	Freyr Bergsteinsson
109	Implement unit test for the add function for products in the config manager	Done	Freyr Bergsteinsson
111	Implement unit test for the update function for products in the config manager	Done	Freyr Bergsteinsson
112	Implement unit test for the delete function for products in the config manager	Done	Gunnar Helgason
113	Implement the add function for products in the config manager	Done	Gunnar Helgason
115	Implement the update function for products in the config manager	Done	Freyr Bergsteinsson
116	Implement the delete function for products in the config manager	Done	Freyr Bergsteinsson
117	Implement unit test for updating with addIfNotExist flag set to true	Done	Freyr Bergsteinsson
26	As a user I want delivery reports to be stored in a database so I can use it to generate a report later	Committed	
86	Implement createNewDelivery function in the config manager that adds a new entry in the delivery report database	Done	Sigurbjorn Kristjansson
78	As an instructor and examiner I want to see the project documents updated with regards to points made in Status Meeting 1	Done	
90	Update risk analysis and split the "work hindered" into more specific risks	Done	Sigurbjorn Kristjansson
91	Update estimated work hours to include additional requirements (300-350 hrs / person)	Done	Gunnar Helgason
74	As an instructor and examiner I want to see a draft of a design document so that I can better understand how the product is supposed to work	Committed	
92	Update sequence diagram for the master script	Done	Gunnar Helgason
93	Create a diagram for script that retrieves available products and their versions	Done	Gunnar Helgason

95	Create a diagram for script that checks if task header dates are ok	Done	Gunnar Helgason
119	Create a database schema for the product table, config table and delivery table	Done	Freyr Bergsteinsson
94	Create a diagram for script that checks if task header numbers (incremental) are ok	Done	Freyr Bergsteinsson

Burndown chart



Sprint 4 (Ironhide), March 9th - March 23nd

Planning

A couple of tasks were left unfinished from Sprint 3 (Megatron) and were added to Sprint 4 (Ironhide). The MoSCoW method was again used to prioritize the user stories and the stories broken down into tasks as with the previous sprint.

User stories

User story	Estimated story points
As a user I want to be able to look at and edit data in an admin web interface to better control the data in a visual manner	1
As a user I want delivery reports to be stored in a database so I can use it to generate a report later	2
As an instructor and examiner I want to see a draft of a design document so that I can better understand how the product is supposed to work	3
As a user I want to fetch a list of projects and versions from Perforce so I can use them to select what to deliver	3
As a developer I want to have a list of auxiliary requirements so that I can continue developing if all other requirements are met	1
As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	1
As an instructor and examiner I want to see a presentation of the product for the second status meeting (Stöðufundur 2) so I can assess the progress of the project	1
As a user I want to automatically get user manuals for a specific version of my software from SharePoint so I can add it to the delivery	3

Tasks

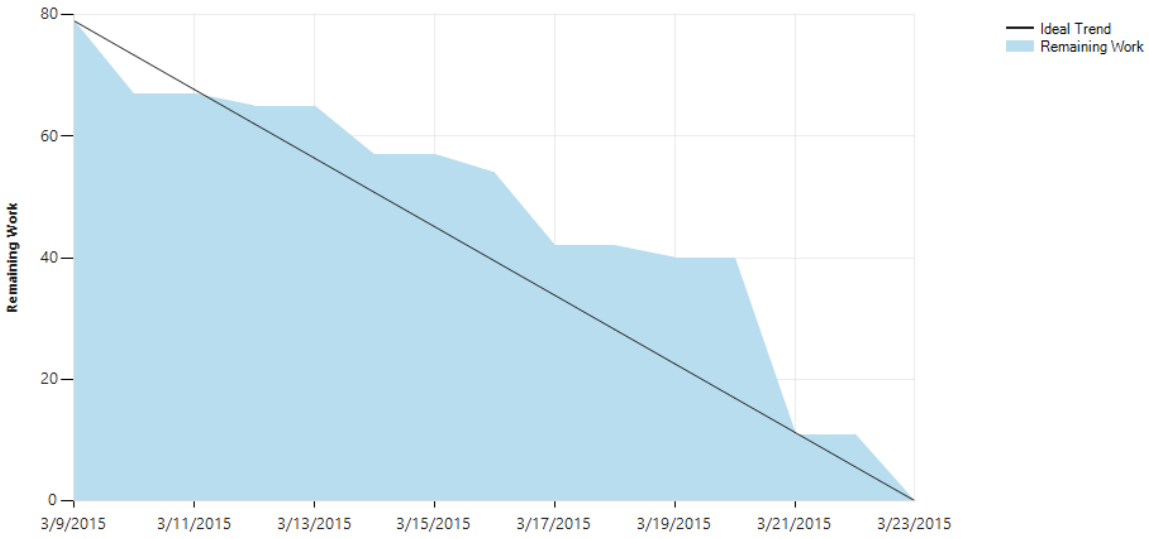
Delivery Automation Tool for Isavia Team - Sprint 4 - Ironhide - Backlog

ID	Title	State	Assigned To	Remaining Work
153	As a user I want to be able to look at and edit data in an admin web interface to better control the data in a visual manner	Committed		
154	Add the current models to the admin interface in the Django project	Done	Árni Þorvaldsson	
26	As a user I want delivery reports to be stored in a database so I can use it to generate a report later	Committed		
108	Implement unit test for the Delivery model	Done	Sigurbjorn Kristjansson	
89	Have the master script generate a new delivery instance	Done	Gunnar Helgason	
74	As an instructor and examiner I want to see a draft of a design document so that I can better understand how the product is supposed to work	Committed		
97	Create a diagram for script that retrieves list of tasks from JIRA/TargetProcess	Done	Freyr Bergsteinsson	
96	Create a diagram for script that checks if all task numbers (and only those task numbers) are in the delivery	Done	Freyr Bergsteinsson	
98	Create a diagram for script that retrieves product build	Done	Freyr Bergsteinsson	
99	Create a diagram for script that retrieves documentation	Done	Gunnar Helgason	
100	Create diagram for script that exports delivery report to PDF	Done	Gunnar Helgason	
101	Create diagram for script that creates the ISO and MD5 files	Done	Gunnar Helgason	
103	Create a diagram for the master script that shows how it handles warnings and errors	Done	Freyr Bergsteinsson	
102	Create diagram for script that copies the ISO, MD5 and PDF files to a storage server	Done	Freyr Bergsteinsson	
156	Create a diagram for script that creates files of new, modified and deleted files.	Done	Freyr Bergsteinsson	

28	As a user I want to fetch a list of projects and versions from Perforce so I can use them to select what to deliver	Committed	
152	Ask Tern system administrator for Perforce credentials that only DATI will be using	Done	Freyr Bergsteinsson
123	Have update_product_versions.py retrieve Perforce credentials from the config manager	Done	Gunnar Helgason
124	Have update_product_versions.py retrieve a list of paths from Perforce to get a list of products	Done	Gunnar Helgason
125	For each path, retrieve a list of product from Perforce	Done	Árni Þorvaldsson
126	Update product database with list of products retrieved from Perforce	Done	Gunnar Helgason
127	For each product, retrieve a list of product versions from Perforce	Done	Gunnar Helgason
128	Update product versions database with list of product versions retrieved from Perforce	Done	Árni Þorvaldsson
129	Have update_product_versions.py return the list of product versions as a string	Done	Árni Þorvaldsson
121	As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	Committed	
133	Update development report (framvinduskýrsla) after sprint 3	Done	Árni Þorvaldsson
69	As an instructor and examiner I want to see a presentation of the product for the second status meeting (Stöðufundur 2) so I can assess the progress of the project	Committed	
122	Add to design document examples of different task headers	Done	Freyr Bergsteinsson
132	Update presentation	Done	Sigurbjorn Kristjansson
136	Update and prepare Project Report (Verkefnislýsing) for status meeting 2	Done	Árni Þorvaldsson
138	Update and prepare Working Arrangement (Verkskipulag) for status meeting 2	Done	Árni Þorvaldsson
139	Update and prepare Project Schedule (Verkáætlun) for status meeting 2	Done	Árni Þorvaldsson
140	Update and prepare Risk Analysis (Áhættugreining) for status meeting 2	Done	Árni Þorvaldsson
141	Create a draft for Operational Manual (Rekstrarhandbók) for status meeting 2	Done	Freyr Bergsteinsson
142	Create a draft for User Guide (Notendaleiðbeiningar) for status meeting 2	Done	Freyr Bergsteinsson

144	Update and prepare Design Document (Hönnunarskýrsla) for status meeting 2	Done	Freyr Bergsteinsson
145	Update and prepare Work Journal (Vinnudagbók) for status meeting 2	Done	Árni Þorvaldsson
31	As a user I want to automatically get user manuals for a specific version of my software from SharePoint so I can add it to the delivery	Committed	
149	Ask Tern system administrator for SharePoint credentials that only DATI will be using	Done	Freyr Bergsteinsson
147	Retrieve SharePoint credentials from the config manager	Done	Árni Þorvaldsson
148	Have script log in to SharePoint with credentials	Done	Freyr Bergsteinsson
150	Retrieve URLs for documents for the given product from the config manager	Done	
151	Retrieve documents form SharePoint with the URLs retrieved form the config manager	Done	

Burndown chart



Sprint 5 (Jazz), March 24rd - April 5th

User Stories

User story	Estimated story points
As a user I want to be able to fetch a build from ParaBuild so that I can include it in the delivery	5
As a user I want the delivery to show up in a list of deliveries (in Django) so I can see what has been delivered	2
As a user I want the delivery output files to be uploaded to a storage server so that I have a backup of the data	3
As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	1
As an instructor and examiner I want to see the project documents updated with regards to points made in Status Meeting 2	1

Tasks

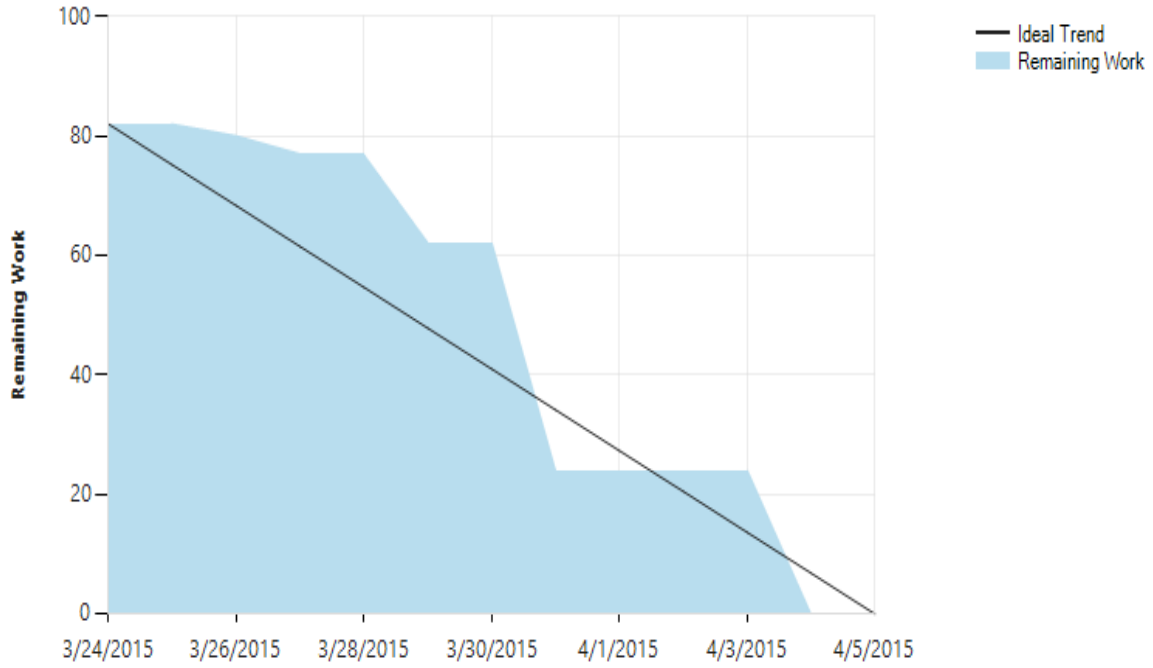
Delivery Automation Tool for Isavia Team - Sprint 5 - Jazz - Backlog

ID	Title	State	Assigned To	Remaining Work
121	As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	Committed		
134	Update Development Report (Framvinduskýrsla) after sprint 4	Done	Árni Þorvaldsson	

158	Update Project Schedule (Verkáætlun) after Sprint 4	Done	Árni Þorvaldsson
77	As an instructor and examiner I want to see the project documents updated with regards to points made in Status Meeting 2	Committed	
146	Create a document that shows the status of the project, which scripts are finished, etc.	Done	Gunnar Helgason
27	As a user I want to be able to fetch a build from ParaBuild so that I can include it in the delivery	Committed	
160	Retrieve the URL for the Parabuild server from Config Manager	Done	Freyr Bergsteinsson
161	Retrieve option whether to include source tarballs from the Config Manager	Done	Freyr Bergsteinsson
162	Retrieve the Staging Directory from the Config Manager	Done	Freyr Bergsteinsson
163	Retrieve URL for build tarballs from Parabuild using the SOAP API	Done	Freyr Bergsteinsson
164	Update the Delivery Description field in the Delivery object after tarballs have been retrieved	Done	Sigurbjorn Kristjansson
165	Retrieve build tarballs from Parabuild and save them to the staging directory	Done	Sigurbjorn Kristjansson
172	Create unit test for the script	Done	Freyr Bergsteinsson
30	As a user I want the delivery to show up in a list of deliveries (in Django) so I can see what has been delivered	Committed	
166	Create a view in Django	Done	Gunnar Helgason
167	Show a list of deliveries in the view	Done	Árni Þorvaldsson

2	As a user I want the delivery output files to be uploaded to a storage server so that I have a backup of the data	Committed	
168	Retrieve hostname, username and password to the storage server from Config Manager	Done	Sigurbjorn Kristjansson
169	Retrieve the Delivery Directory from the Config Manager	Done	Sigurbjorn Kristjansson
170	Upload files in the delivery directory to the storage host using the server credentials	Done	Freyr Bergsteinsson
171	Create a unit test for the script	Done	Freyr Bergsteinsson

Burndown chart



Sprint 6 (Optimus Prime), April 6th - April 19th

User Stories

User story	Estimated story points
As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	1
As a developer I want a parsing tool to recognize task headers so that it becomes easier to examine them in scripts	5
As a developer I want to connect to TargetProcess through an API to get access to task numbers	4
As a user I want to make sure that task header numbers are correctly numbered in an incremental fashion so that the delivery isn't rejected	2

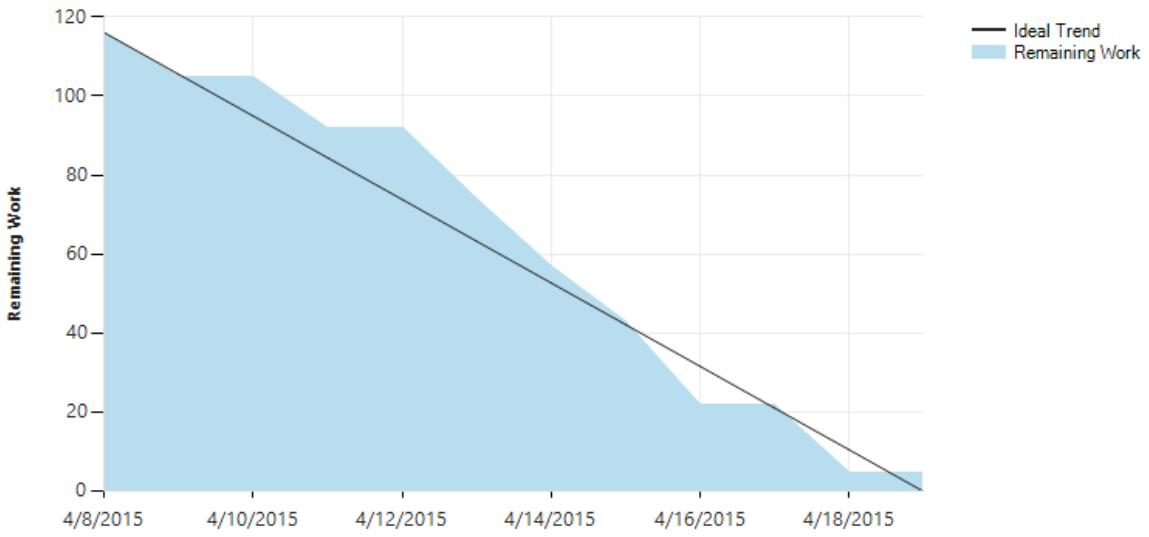
Tasks

ID	Title	State	Assigned To	Remaining Work
121	As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	Done		
174	Update Development Report (Framvinduskýrsla) after sprint 5	Done	Árni Þorvaldsson	
178	Update Project Schedule (Verkáætlun) after Sprint 5	Done	Árni Þorvaldsson	

157	As a developer I want a parsing tool to recognize task headers so that it becomes easier to examine them in scripts	Done	
182	Design exactly how the parser should work	Done	Freyr Bergsteinsson
194	Implement interface for the FileParser module	Done	Freyr Bergsteinsson
195	Create unit tests for the parse function in FileParser	Done	Freyr Bergsteinsson
196	Get actual task headers from Tern code base to use for unit testing	Done	Freyr Bergsteinsson
197	Create RegEx for first line of Task Header	Done	Sigurbjorn Kristjansson
199	Implement parse_date function	Done	Freyr Bergsteinsson
200	Implement parse function which returns a list of TaskHeaderEntry objects	Done	Sigurbjorn Kristjansson
201	Create unit tests for the parse_date function in the FileParser module	Done	Freyr Bergsteinsson
173	As a developer I want to connect to TargetProcess through an API to get access to task numbers	Done	
183	Investigate TargetProcess API	Done	Árni Þorvaldsson
184	Connect to Tern's TargetProcess site with credentials from the Config Manager	Done	Gunnar Helgason
185	Add TargetProcess credentials to Config Manager	Done	Gunnar Helgason
186	Get information about a specific Product Release from TargetProcess, using the ProductVersion of a delivery	Done	Sigurbjorn Kristjansson
187	Get list of task numbers for the given product version	Done	Sigurbjorn Kristjansson

24	As a user I want to make sure that task header numbers are correctly numbered in an incremental fashion so that the delivery isn't rejected	Done	
188	Get a list of new and modified files since the previous release	Done	Freyr Bergsteinsson
189	Parse the new and modified files with the File Parser	Done	
190	Check validity of task headers entry numbers of task headers returned from File Parser	Done	
191	Investigate how to get a list of new and modified files from Perforce	Done	Freyr Bergsteinsson
192	Investigate how to get the contents of a file from Perforce	Done	Freyr Bergsteinsson
193	Investigate how to get the difference of a file from previous delivery	Done	Freyr Bergsteinsson

Burndown chart



(A completed task's status was not set as done, therefore the chart shows remaining work)

April 21st - April 26th: 6 days time off due to exams and family obligations

Sprint 7 (Hot Rod), April 27th - May 3rd

User stories

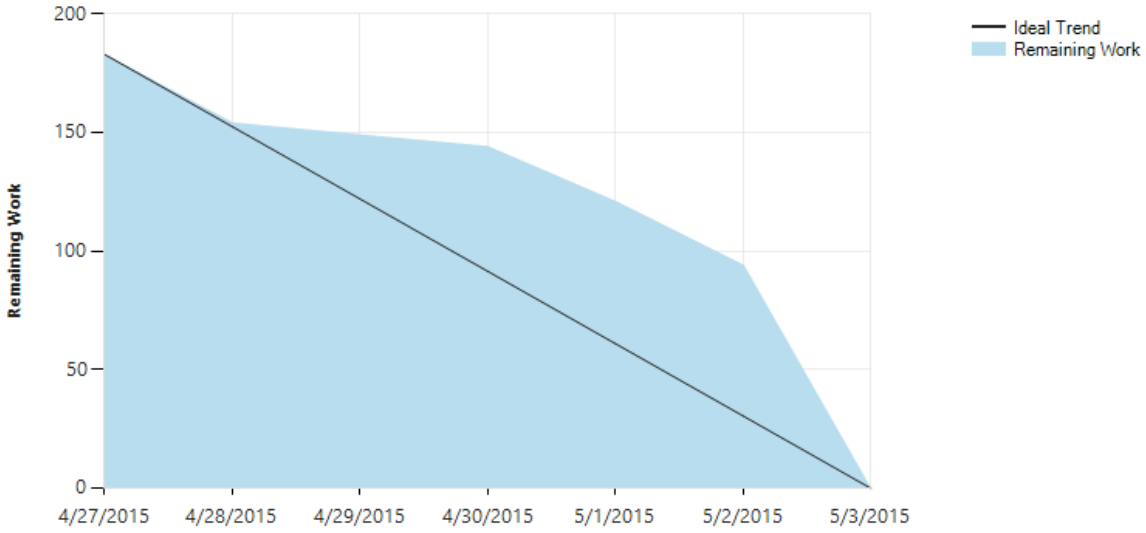
User story	Estimated story points
As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	1
As a user I want to make sure that the task headers delivered contain valid task numbers so that the delivery isn't rejected	3
As a user I want task headers to show the correct date in task headers so that the delivery isn't rejected	5
As a user I want to retrieve all documents from a folder in SharePoint for a product instead of one document	3
As a user I don't want old backup files to be overwritten in case of mistakes in new delivery	3
As an instructor and examiner I want to see a presentation of the product for the third status meeting (Stöðufundur 3) so I can assess the progress of the project	1
As a user I want to be able to create a new delivery with a command line script or perform re-delivery on an existing delivery	3

Tasks

ID	Title	State	Assigned To	Remaining Work
121	As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	Committed		
175	Update Development Report (Framvinduskýrsla) after sprint 6	Done	Árni Þorvaldsson	
179	Update Project Schedule (Verkáætlun) after Sprint 6	Done	Árni Þorvaldsson	
215	Update risk analysis because of risk of using cloud based service	Done	Árni Þorvaldsson	
20	As a user I want to make sure that the task headers delivered contain valid task numbers so that the delivery isn't rejected	Committed		
202	Get task numbers in current delivery from the Delivery object	Done	Sigurbjorn Kristjansson	
203	Get list of new and modified files in this delivery compared to last delivery	Done	Sigurbjorn Kristjansson	
204	Parse task headers in new and modified files	Done	Sigurbjorn Kristjansson	
205	Go through all new task header entries and compare to list of task numbers in delivery	Done	Sigurbjorn Kristjansson	
18	As a user I want task headers to show the correct date in task headers so that the delivery isn't rejected	Committed		
208	Get list of new and modified files in this delivery compared to last delivery	Done	Freyr Bergsteinsson	
209	Parse task headers in new and modified files	Done	Freyr Bergsteinsson	
210	Go through all new task header entries and compare date format to date format in previous task header entry	Done	Freyr Bergsteinsson	
211	Make sure dates in new task header entries are not in the future	Done	Freyr Bergsteinsson	

212	Make sure the date in new task header entries are not older than the date of the previous delivery	Done	Freyr Bergsteinsson
226	Create unit tests for check_task_header_dates script	Done	Freyr Bergsteinsson
214	As a user I want to retrieve all documents from a folder in SharePoint for a product instead of one document	Committed	
222	Have a config value for the directory of documents for each product with a placeholder for the product version	Done	Árni Þorvaldsson
223	Update script to iterate through folder instead of using url for one file and download each file in the folder	Done	Gunnar Helgason
213	As a user I don't want old backup files to be overwritten in case of mistakes in new delivery	Committed	
216	Investigate transactions of large files when copying to storage server	Done	Gunnar Helgason
221	Rename existing folder to a unique name before uploading delivery directory	Done	Gunnar Helgason
70	As an instructor and examiner I want to see a presentation of the product for the third status meeting (Stöðufundur 3) so I can assess the progress of the project	Committed	
218	Update operational manual, and user guide	Done	Freyr Bergsteinsson
219	Review other documents and update accordingly	Done	Árni Þorvaldsson
217	As a user I want to be able to create a new delivery with a command line script or perform re-delivery on an existing delivery	Committed	
224	Update create delivery script	Done	Freyr Bergsteinsson

Burndown chart



Sprint 8 (Mirage), May 4th - May 10th

User stories

User story	Estimated story points
As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	1
As a user I want to make sure that the task headers delivered contain valid task numbers so that the delivery isn't rejected	3
As an instructor and examiner I want to see a presentation of the product for the third status meeting (Stöðufundur 3) so I can assess the progress of the project	1
As a user I want to view one specific delivery report so I can verify its correctness	3
As a user I want to generate a delivery report in PDF format so that I can add it to the delivery	3
As an instructor and examiner I want to see the project documents updated with regards to points made in Status Meeting 3	1
As a user I want the run_script script to behave the same as the Script.execute method	1
As a user I don't need config values for a specific product version (only for products)	1
As a user I want the system to work on product versions regardless of minor version being zero padded or not	3

As a user I want the system to automatically get a list of new, modified and removed files between to deliveries	4
As a user I want the system to automatically create an ISO of a staging directory and MD5	3

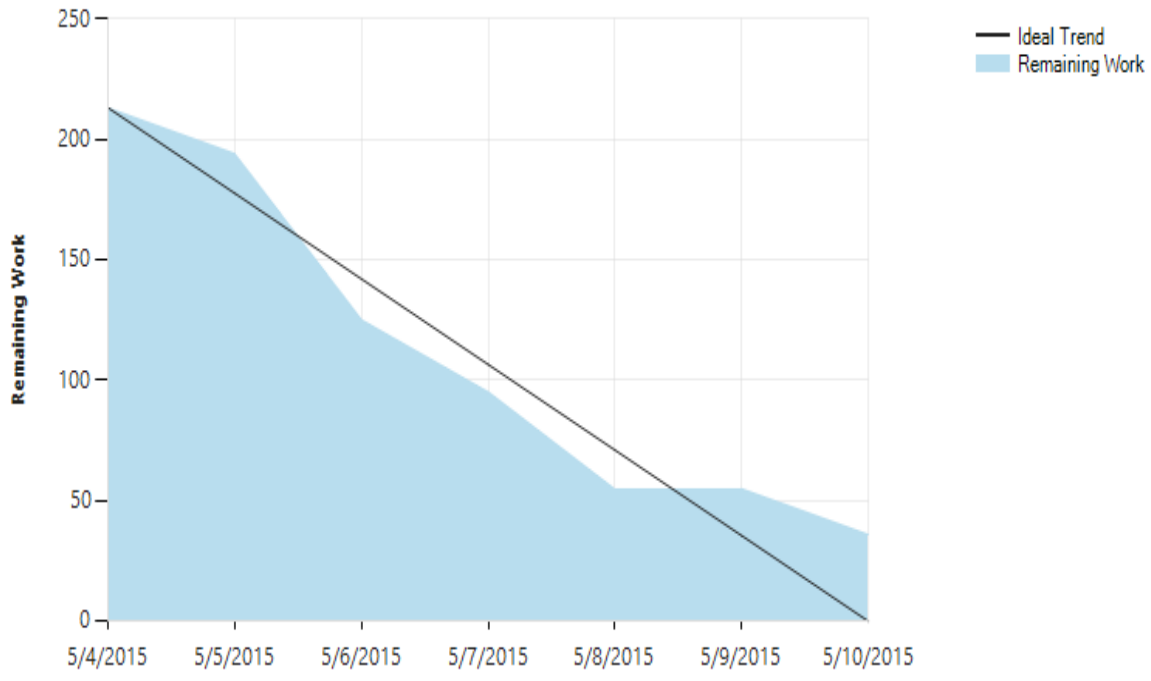
Tasks

ID	Title	State	Assigned To	Remaining Work
121	As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	Committed		
176	Update Development Report (Framvinduskýrsla) after sprint 7	To Do		2
180	Update Project Schedule (Verkáætlun) after Sprint 7	To Do		2
231	Email all documents to instructor and examiner	To Do		1
20	As a user I want to make sure that the task headers delivered contain valid task numbers so that the delivery isn't rejected	Committed		
206	Make sure the task numbers in the task header entries are task numbers in the delivery	In Progress	Sigurbjorn Kristjansson	2
207	Write unit test for the script	In Progress	Sigurbjorn Kristjansson	15
70	As an instructor and examiner I want to see a presentation of the product for the third status meeting (Stöðufundur 3) so I can assess the progress of the project	Committed		
220	Create a presentation for Status Meeting 3	To Do		20
225	Rehearse presentation	To Do		20

232	Do a retrospective meeting after Status Meeting 3	To Do	1
227	As a user I want to view one specific delivery report so I can verify its correctness	Committed	
233	Create a view in Django for viewing of single delivery reports	To Do	25
19	As a user I want to generate a delivery report in PDF format so that I can add it to the delivery	Committed	
234	Install and investigate how to use PDFKit	To Do	2
235	Create PDF from view of a specific delivery report	To Do	10
236	Store PDF of delivery document in Delivery Directory	To Do	5
76	As an instructor and examiner I want to see the project documents updated with regards to points made in Status Meeting 3	Committed	
237	Placeholder for things to do with regards to points made in status meeting 3	To Do	20
230	As a user I want the run_script script to behave the same as the Script.execute method	Committed	
238	Update run_script.py script	To Do	5
229	As a user I don't need config values for a specific product version (only for products)	Committed	
239	Remove ProductVersion from Config model	To Do	5
228	As a user I want the system to work on product versions regardless of minor version being zero padded or not	Committed	
240	Update retrieve_document_from_sharepoint to use zero-padded minor version	To Do	10
241	Investigate the need to update other scripts w.r.t. zero padded minor version	To Do	10

243	As a user I want the system to automatically get a list of new, modified and removed files between to deliveries	Committed	
244	Create unit test for script	To Do	15
245	Read data from perforce to create files	To Do	15
246	Store files in staging directory	To Do	2
242	As a user I want the system to automatically create an ISO of a staging directory and MD5	Committed	
247	Create unit tests for script	To Do	15
248	Create an ISO image of the staging directory	To Do	10
249	Store ISO image and MD5 checksum of it in the delivery directory	To Do	5

Burndown chart



Sprint 9 (Ultra Magnus), May 11th - May 17th

User stories

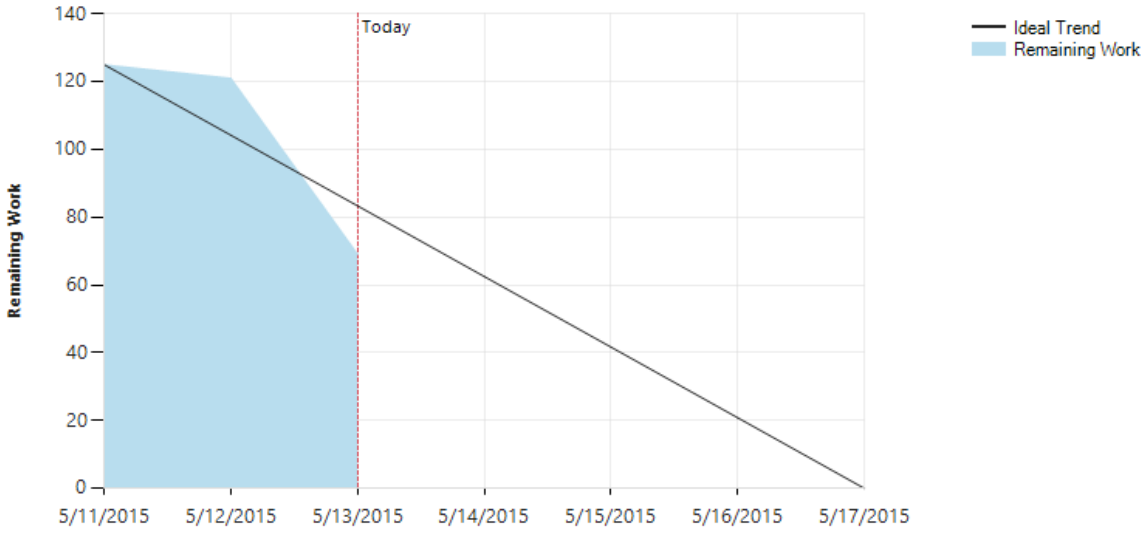
User story	Estimated story points
As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	1
As an instructor and examiner I want to see the project documents updated with regards to points made in Status Meeting 3 (unfinished from previous sprint)	1
As an instructor and examiner I want to see a final report for the project so I can grade the project	8

Tasks

ID	Title	State	Assigned To	Remaining Work
260	As an instructor and examiner I want to see a presentation of the product for the final presentation	Committed		
261	Create a presentation	To Do		20
262	Rehearse presentation	To Do		20
121	As an instructor and examiner I want to see the work progress updated regularly to have a better overview of the project	Committed		
177	Update Development Report (Framvinduskýrsla) after sprint 8	Done		
181	Update Project Schedule (Verkáætlun) after Sprint 8	Done		

76	As an instructor and examiner I want to see the project documents updated with regards to points made in Status Meeting 3	Committed		
237	Investigate how to perform deployment of DATI to production server	In Progress	Gunnar Helgason	20
250	Update operational manual to include deployment instructions	Done	Freyr Bergsteinsson	
253	Get unit test coverage up to 90%	Done	Freyr Bergsteinsson	
254	Test setup on fresh computer	Done	Freyr Bergsteinsson	
71	As an instructor and examiner I want to see a final report for the project so I can grade the project	Committed		
255	Finish the introduction section of the final report	In Progress		4
256	Finish the working arrangements of the final report	Done		
257	Finish the progress section of the final report	Done		
258	Finish the conclusion section of the final report	In Progress		4
259	Finish the appendices of the final report	Done		
263	Send the report to the printer BEFORE wednesday 15:00	To Do		1

Burndown chart



HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

Tern Systems - Risk Analysis (Áhættugreining)

T-404-LOKA

Students:
Árni Þorvaldsson (arnitho12@ru.is)
Freyr Bergsteinsson (freyrb12@ru.is)
Gunnar Þór Helgason (gunnarh07@ru.is)
Sigurbjörn Kristjánsson (sigurbjorn12@ru.is)

Instructor:
Daníel Máni Jónsson (daniel@valitor.is)

Examiner:
Hannes Pétursson (hap@ru.is)

Table of Contents

Document History

Date	Author	Description
2015-02-09	ÁÞ, FB, GÞH, SK	Initial version.
2015-02-25	ÁÞ, FB, GÞH, SK	“Work capability hindered” split into more precise events.
2015-04-25	ÁÞ	External risks added.

Risk Analysis

This risk analysis is our careful examination of anything that can affect the Delivery Automation Tool for ISAVIA. The risk analysis is divided into two parts, internal and external risks. Internal risks contains things that might affect the group and its members that don't have to do with the actual product. External risks are things that might affect the technical side of the project.

Probability and effect legend

Table 0. Probability and effect of occurrences

Points	Probability of occurrence	Effect of occurrence
5 - Severe	Very Likely	Massive impact
4 - Significant	Likely	Major impact
3 - Moderate	Neither nor	Moderate impact
2 - Minor	Unlikely	Minor impact
1 - Minimal	Very Unlikely	Slight impact

Internal risk

Table 1. Internal risk

Risk/Hazard	Probability	Effect severity	Risk factor	Preventive measure	Solution to Risk
Difference in work contributions	4	2	8	Assign manageable tasks	Pair on tasks, ask questions and share information
Design flaws	2	4	8	Communicate often with PO	Revise design
Work/School obligations	4	2	8	Organize time	Make up for lost time with additional work contribution
Sprint time plan does not meet the schedule	3	2	6	Keep constant workflow	Learn from previous iterations
Illness within the group	3	2	6	Take your vitamins	Reevaluate sprint capacity
Lack of knowledge of tools in use*	3	2	6	Read documentation	Go through tutorials
Close relative or friend passes away	1	5	5	N/A	Show support and reevaluate sprint capacity
Project release does not meet the deadline	2	2	4	Keep constant workflow	Deliver project documents to Tern for further development
Lack of interest within the group	1	4	4	Assign manageable tasks	Mention problem in retrospective meetings
Transportation difficulties	4	1	4	Carpool, car maintenance	Carpool, use bus
Communication difficulty within group	1	3	3	Daily standup meetings	Mention problem in retrospective meetings
Task definition not good enough*	1	3	3	Prepare tasks thoroughly	Talk with PO
Family obligations	2	1	2	Organize time, if possible	Make up for lost time with additional work contribution

External risk

Table 2. External risk

Risk	Probability	Effect severity	Risk factor	Preventive measure	Solution to Risk
Time spent on using unsuitable frameworks	3	2	6	Investigate framework usability	Consult PO of framework choices
Technical problems, e.g., the data is lost	1	5	5	Push frequently to github	Pull from github
Time spent on talking about movies and other stuff on meetings (can't keep focus)	5	1	5	Be up-to-date on EVERYTHING	Get to the point quickly
Cloud based services (Git, Visual Studio Online) becoming unavailable long term	1	4	4	Keep backup of data (when available and/or practical)	Change to another cloud based service
System changes at Tern System	4	1	4	Assume there will be changes	Configurable solutions
Lack of information	3	1	3	Schedule regular meetings with PO	Communicate more often with PO
Cloud based services (Git, Visual Studio Online) becoming unavailable short term	2	1	2	Keep backup of data (when available and/or practical)	Use backup data while service is down
Workplace at Tern Systems becomes unavailable	2	1	2	Book conference room in advance	Find another location
Communication difficulty with the company	2	1	2	Clearly outline goals and status within project	Schedule meeting with PO
Communication difficulty with the instructor	2	1	2	Clearly outline goals and status within project	Schedule meeting with instructor

Computer fails within the group	1	1	1	Version control	Receive a working computer from Tern

Main Risks

In our mind, the main risks are:

- Difference in work contribution.
- Work/school obligations
- Design flaws.

Difference in work contributions and work or school obligation are maybe related a little, because work/school obligations may interfere with how much work contributions a group member can offer. If a big assignment is due and the group member has underestimated the time he will have to spent to his obligation in the school or work the time available to work on on other tasks as the automation tool. We think of this is the main risk since it's more likely to occur than the other 2 who have the same risk factor.

Design flaws have also a big risk factor because we are working on new things in an unfamiliar territories. We have already change some feature of the system that we had designed because we simply found a better way to the design the project. And for certain a redesign is more likely to occur rather than not thought out the project.

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

Tern Systems - Operational Manual (Rekstrarhandbók)

T-404-LOKA

Students:
Árni Þorvaldsson (arnitho12@ru.is)
Freyr Bergsteinsson (freyrb12@ru.is)
Gunnar Þór Helgason (gunnarh07@ru.is)
Sigurbjörn Kristjánsson (sigurbjorn12@ru.is)

Instructor:
Daníel Máni Jónsson (daniel@valitor.is)

Examiner:
Hannes Pétursson (hap@ru.is)

Table of Contents

Document History

Date	Author	Description
2015-03-19	FB	Initial version.
2015-05-12	FB	Various updates.

System Requirements

The Delivery Automation Tool for Isavia (DATI) tries to be a system architecture-independant as possible. That being said, it is intended to run on any Linux distribution using kernel version 2.6 or above. It will also run fine on OSX and with only minor adjustments it will also run on Windows (see Appendix A for details).

Operating System	Linux 2.6+ (Fedora, CentOS, Ubuntu, etc.) OSX 10.6+
CPU type	**For 32-bit systems:** 32-bit Intel® Pentium® 4 or AMD Athlon™ Dual Core **For 64-bit systems:** AMD Athlon 64 AMD Opteron™ Intel® Xeon® with Intel EM64T support Intel Pentium 4 with Intel EM64T support
Network	NIC with Ethernet 10/100/1000 Cat 5/6
Memory	1 GB (2 GB recommended)
Display	N/A (will run on a headless setup)
Disk Space	1 GB

DATI will also run on a virtual machine emulating these hardware requirements.

System Installation

This chapter describes how to install the system that will run DATI. If you already have a running computer with all the system-wide software installed (e.g. MySQL and Python - see below for details), the you can jump directly to the chapter Product Installation.

Operating System

We recommend installing the 64-bit version of Fedora 21, although any Linux 2.6+ operating system will do. You can download the ISO for Fedora 21 here.

You can burn the ISO to a DVD or to a bootable USB flash drive and then boot the computer to install the system on using that device. If you are installing on a virtual machine, use the ISO as an optical drive for that machine.

Follow the instructions on screen; all default settings are acceptable. Once the installation is complete and the computer is network connected, you can remove the screen (for a non-virtual installation) and connect via network. Just make sure that the SSH server is running before you do - as root, run the following commands:

```
# systemctl start sshd.service
# systemctl enable sshd
```

The first line starts the SSH server if it wasn't already running and the second line registers the SSH server to be started on system startup (i.e. after reboot).

We recommend creating a new user dedicated for installing DATI. To create a user called *dati*, run the following commands as root:

```
# useradd dati
# passwd dati
```

When prompted for a password for that user, be aware of security issues regarding password creation. Use a good password.

The rest of this document assumes that DATI will be installed and run as this user.

System-wide software packages

MySQL

DATI requires MySQL to be installed. As root, run the commands

```
# yum install mariadb-server
# systemctl start mysqld.service
# systemctl enable mysqld.service
```

Note: MariaDB is a fork of MySQL and is identical to MySQL, but is licensed under GNU GPL.

Python

DATI requires Python 2.7, which comes pre-installed with Fedora 21. If you are using another operating system and Python 2.7 does not comes pre-installed, then you can use the software package manager for that operating system to install Python 2.7. DATI does currently not support Python 3.

Make sure that pip - the package installer for Python - is installed on the system:

```
# yum install python-pip
```

Other

A few more packages are needed for various tasks. These can all be installed with yum:

```
# yum install libffi-devel genisoimage mariadb-devel \
              gcc-c++ python-devel
```

- `libffi-devel` is used to allow SSL connections for both SharePoint and Perforce.
- `genisoimage` installs the program mkisofs among other things. The program mkisofs is used to create ISO images, which is used by DATI.
- `mariadb-devel` is used by the mysql-connector-python package.
- `gcc-c++` is the GNU C/C++ compiler. It is used to install various python packages.
- `python-devel` is used by the MySQL-python package.

Virtual Environment

The Python VirtualEnv package will help create isolated environments, which help keep the global site-packages directory clean and manageable. It also eases the transition when moving DATI between different computers.

As root, run the following command:

```
# pip install virtualenv
```

Once the virtualenv package has been installed on the system, create a new virtual environment in the directory where DATI will be installed. As the user *dati*, run the following command in dati's home directory:

```
$ virtualenv .venv
```

This creates a hidden directory named *.venv* in the current directory. To switch to the virtual environment, run the following command:

```
$ source .venv/bin/activate
```

After running this command you will have switched to the virtual environment, which is indicated by the name of the virtual environment in parenthesis at the beginning of the command prompt. Having switched to this virtual environment you will be able to install python packages without interfering with the system's Python packages.

To switch back to the system environment, run the following command:

```
(.venv)$ deactivate
```

Read more about virtual environments in Python here.

Firewall settings

DATI runs a web server that listens to port 8000 by default. You can configure this later (see chapter Running the server), but you need to make sure that the firewall on the server allows external connections to this port.

Since it is assumed that the server will be running on an internal network, we suggest turning off the firewall software that comes installed with Fedora. Run the following command as root:

```
# systemctl stop firewalld.service
# systemctl disable firewalld.service
```

If you are running another operating system, then the exact command to turn off the firewall may differ. If you wish to keep the firewall running and only open for the port that the DATI web server is running on, then refer to the documentation for the firewall software.

Product Installation

This section describes how to install DATI, given that an operating system and all system dependencies have already been installed.

Code Base

> NOTE: At the time of this writing, how the code for DATI is retrieved is described as it is done during development. In the future, after DATI has been delivered to Tern Systems, how the code for DATI is installed on the system may differ from what is described here.

The code for DATI is stored in a github repository which can be retrieved by cloning it:

```
$ git clone https://github.com/arnitho/loka-tern
```

This is a private repository. If wish to gain access, please send an email to arnitho12@ru.is.

Python Packages

DATI uses various python packages. They can be installed all at once in the virtual environment using the file requirements.txt in the install directory of the DATI root directory. In the virtual environment run the following command:

```
(.venv)$ install/install_python_packages.sh
```

Product Database

Once MySQL has been installed, the database used by DATI can be initialized with the following command:

```
(.venv)$ script/initialize_database.sh
```

This script will create a new database called isavia_deliveries and a database user named django (same password as username). It will also create tables as they are generated by Django from the defined models in the project. Finally, it will populate the tables with default values.

The default values provided are only for convenience and need to be modified for the installed system to correctly represent the server URLs, user names and other configuration values (see Django administration interface below)

Development

This section explains the operations needed for further development of the project.

Development Web Server

> NOTE: At the time of this writing, DATI is served using a development web server provided by Django. It is not supposed to be deployed to production. In the future, DATI will be deployed to a production server, such as Apache with mod_wsgi or something similar. Until then, this section describes how to get the development server up and running.

To run the web server for DATI, go to the product root directory and run the following command:

```
$ scripts/run_server.sh
```

The web server is configured to run on port 8000 by default. If you wish to change this you can run the script with another port number as its parameter:

```
$ scripts/run_server.sh 2345
```

If you wish for the server to run automatically on system startup, then add the following line to the /etc/rc.local:

```
<path to DATI>/scripts/run_server.sh &
```

If /etc/rc.local does not exist, you need to create it:

```
# touch /etc/rc.local
# chmod +x /etc/rc.local
```

Make sure that the first line in /etc/rc.local includes the bash interpreter:

```
#!/bin/bash
```

Finally, make sure that the service that executes rc.local on system startup is configured:

```
# systemctl enable rc-local.service
```

NOTE: All of the above is for Fedora 21. If you use another operating system you may need to run other commands to make sure the server is run on startup. Refer to the documentation for your operating system.

Django admin interface

Django has a built-in administration web interface that can be used to manually create, modify or delete data in the database. A superuser needs to be defined before the interface can be used. Create the superuser with the following command:

```
$ python dati/manage.py createsuperuser
```

Enter the desired username, email and password for the superuser, then direct your browser to the following URL to log in to the admin interface:

```
http://localhost:8000/admin
```

Unit Tests

All of the code in the project is and should be unit tested, preferably using test driven development. Django has a built-in unit test framework which is used by the project.

To create a unit test for a model or script, create a file called `test_<module>.py` in any of the project's app directories, where <module> describes what part is being tested. A unit test file must contain the following:

- A class that inherits the `django.tests.TestCase` class. The name of the class should be descriptive of what is being tested, e.g. `ProductVersionTest`.
- A `setUp` method inside the class which initializes test data. This method is automatically called before each test method is called. In some cases you may want to define a `tearDown` method, which is automatically called after each test method is called, but usually this is not needed.

- Any number of methods that have the prefix `test_` in the method name, e.g. `test_add_existing_product_version`.

To run all the unit tests in the project, run the following command in the *dati* directory:

```
$ python manage.py test
```

To run all the unit tests for a specific app:

```
$ python manage.py test <app>
```

where <app> is the name of the project app. For example:

```
$ python manage.py test products
```

To run a specific unit test method:

```
$ python manage.py test \
      <app>.test_<module>.<class>.test_<method>
```

where <app> is the name of the project app, <module> is the test module used, <class> is the name of the class that inherits from TestCase and <method> is the specific test you wish to run. For example:

```
$ python manage.py test \
     products.test_product.ProductTest.test_add_new_product
```

Code Coverage

To keep track of how much code is actually tested by unit tests, it is a good idea to run a code coverage tool. DATI comes installed with such a tool. To run the code coverage task:

```
$ scripts/code_coverage.sh
```

You are given a summary of the output on the terminal. To view the code coverage in more detail, point your browser to:

```
http://localhost:8000/coverage
```

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

Tern Systems - User Guide (Notendaleiðbeiningar)

T-404-LOKA

Students:
Árni Þorvaldsson (arnitho12@ru.is)
Freyr Bergsteinsson (freyrb12@ru.is)
Gunnar Þór Helgason (gunnarh07@ru.is)
Sigurbjörn Kristjánsson (sigurbjorn12@ru.is)

Instructor:
Daníel Máni Jónsson (daniel@valitor.is)

Examiner:
Hannes Pétursson (hap@ru.is)

Table of Contents

Document History

Date	Author	Description
2015-03-21	FB	Initial version.
2015-04-28	FB	Updates to usage and viewing of deliveries.

Installation

Please refer to the Operations Manual on how to install the Delivery Automation Tool for Isavia (DATI). Make sure the web server for DATI is running before continuing. The following chapters assume the web server is running on the server *deliveries.tern.is* on port *8000*. See the Operations Manual for how to configure this.

Configuration

DATI communicates with a number of other services on the network. Make sure that server names, credentials and other settings are correctly configured. Review your configuration values by pointing your browser to

```
http://deliveries.tern.is:8000/config
```

If you see that some values are incorrect or missing, then you need to log in to the administration page with the superuser to make the changes:

```
http://deliveries.tern.is:8000/admin/configmanager/config/
```

Viewing Deliveries

All the deliveries in the system can be viewed by pointing your browser to

```
http://deliveries.tern.is:8000/deliveries
```

To view a specific delivery, click on it in the list above, or point your browser to

```
http://deliveries.tern.is:8000/deliveries/<n>
```

where <n> is the number of the delivery.

Creating a New Delivery

There are three ways to create a new delivery:

- On the web for DATI
- On the command line
- On the admin web interface for DATI

The first two in the list invoke the automation process, while the third one overwrites all automation and is used to manually create a delivery.

Using the web interface

> Note: This interface has not been implemented yet.

Point your browser to

```
http://deliveries.tern.is:8000/newdelivery
```

You will be presented with a web page where you can select from a list which product to create a delivery for. When a product has been chosen you can select which version for that product to create a delivery for (newest version selected by default).

Using the command line

For convenience, you can create a new delivery from the command line. Log in to the server running DATI with SSH and navigate to the DATI directory. From there, run the command

```
$ scripts/create_delivery.sh <product> <version>
```

where <product> is the name of the product to create a delivery for, and <version> is the version string of the product to create a delivery for. Example:

```
$ scripts/create_delivery.sh ISDS 15.1
```

Using the admin interface

In case the automation process provided for the previous methods fails (such as an outage of one of the servers used to communicate with), we can always fall back to using a manual method for creating a new delivery. Log in to the administration page with the superuser and point your browser to

```
http://deliveries.tern.is:8000/admin/deliveries/delivery/add/
```

Creating New Scripts

When creating a new script, add it to the directory

```
dati/scripts/collection/
```

The script can contain any valid Python code. It also runs inside the Django project environment, meaning that it can import and access all of the models used. Adding the script in this directory will automatically make it available for the system.

The script must contain at least one function named `run` that takes the delivery number as its only parameter.

To run a single script (convenient when testing), use the supplied run_script wrapper script:

```
$ scripts/run_script.sh <script name> <delivery nr>
```

where <script name> is the name of the script and <delivery nr> is the delivery number, which will be passed as an argument to the script. For example:

```
$ scripts/run_script.sh retrieve_build_from_parabuild 420
```

If a script does not exit with an exit code of 0, then this indicates a fatal error occurred. No further scripts, should there be any, will be run, and any error messages are printed to the standard error.

If a script exits with an exit code of 0, then this indicates no error occurred. Any messages written to standard error will be added to the warning_messages field in the delivery object. Anything written to standard output is ignored, regardless of the exit code.

Selecting Scripts for a Product

For the time being, adding, modifying or removing which scripts should be executed for a certain product must be done in the Django admin interface:

```
http://deliveries.tern.is:8000/admin/scripts/productscript/
```

You have to manually keep track of the sequence numbers for the scripts used - if you are inserting a script in the middle of a sequence, then you need to increase the sequence

numbers of all the scripts executed after the inserted script. To prevent this from happening, you could assign sequence numbers that increase by 10. This way, if you ever need to insert a script in a sequence, you could give it a sequence number between the two sequence numbers of the enclosing scripts. As an example, inserting a script between two scripts that have sequence numbers 10 and 20 should use a sequence number of 15.

In the future, there will be a nice web interface which automatically takes care of the sequence numbers, and the GUI will consist of just a list of the chosen script names that can be reordered visually.