FORGE Project Planning Overview

Pasi Kivikangas

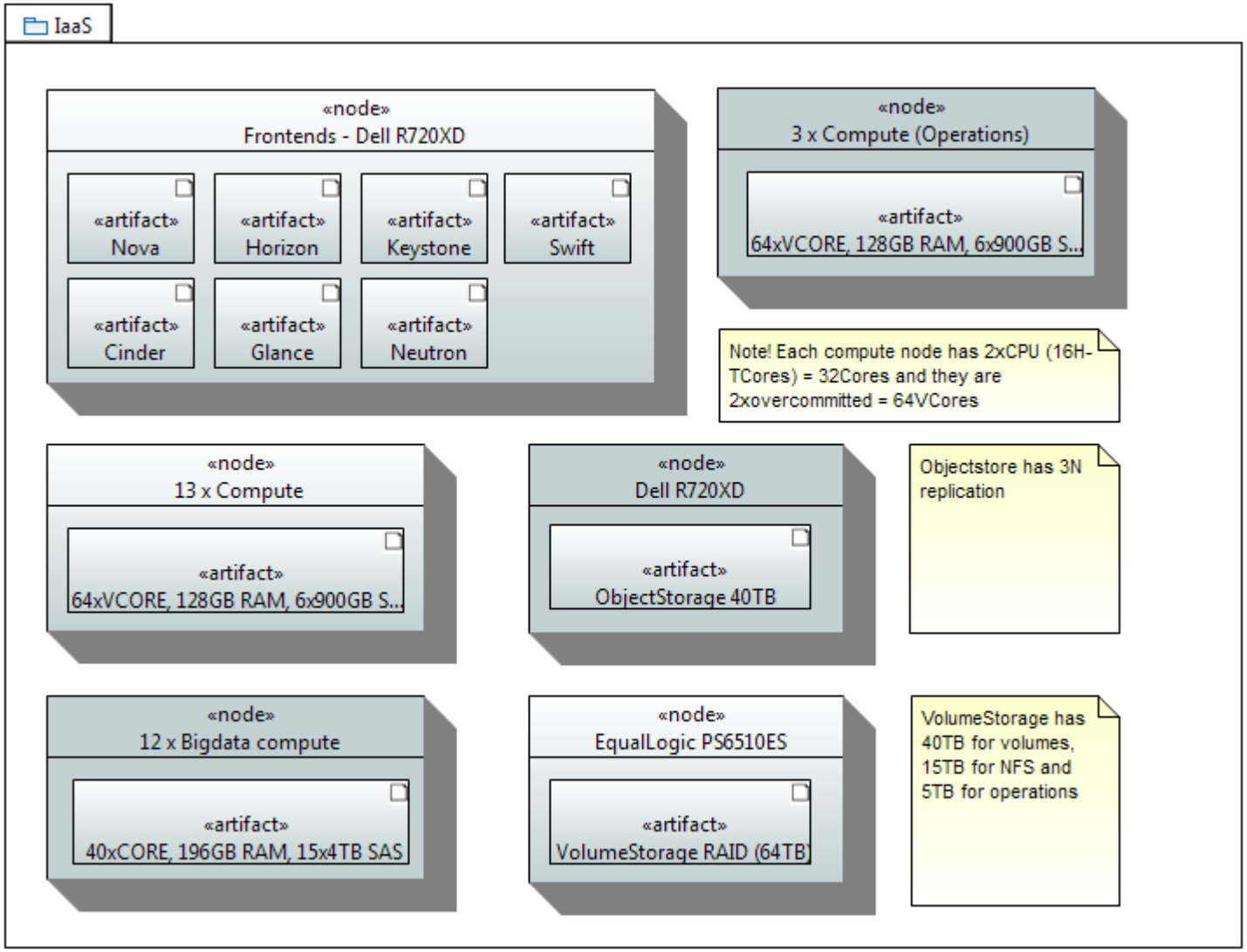
Project Manager

DIGILE Ltd.

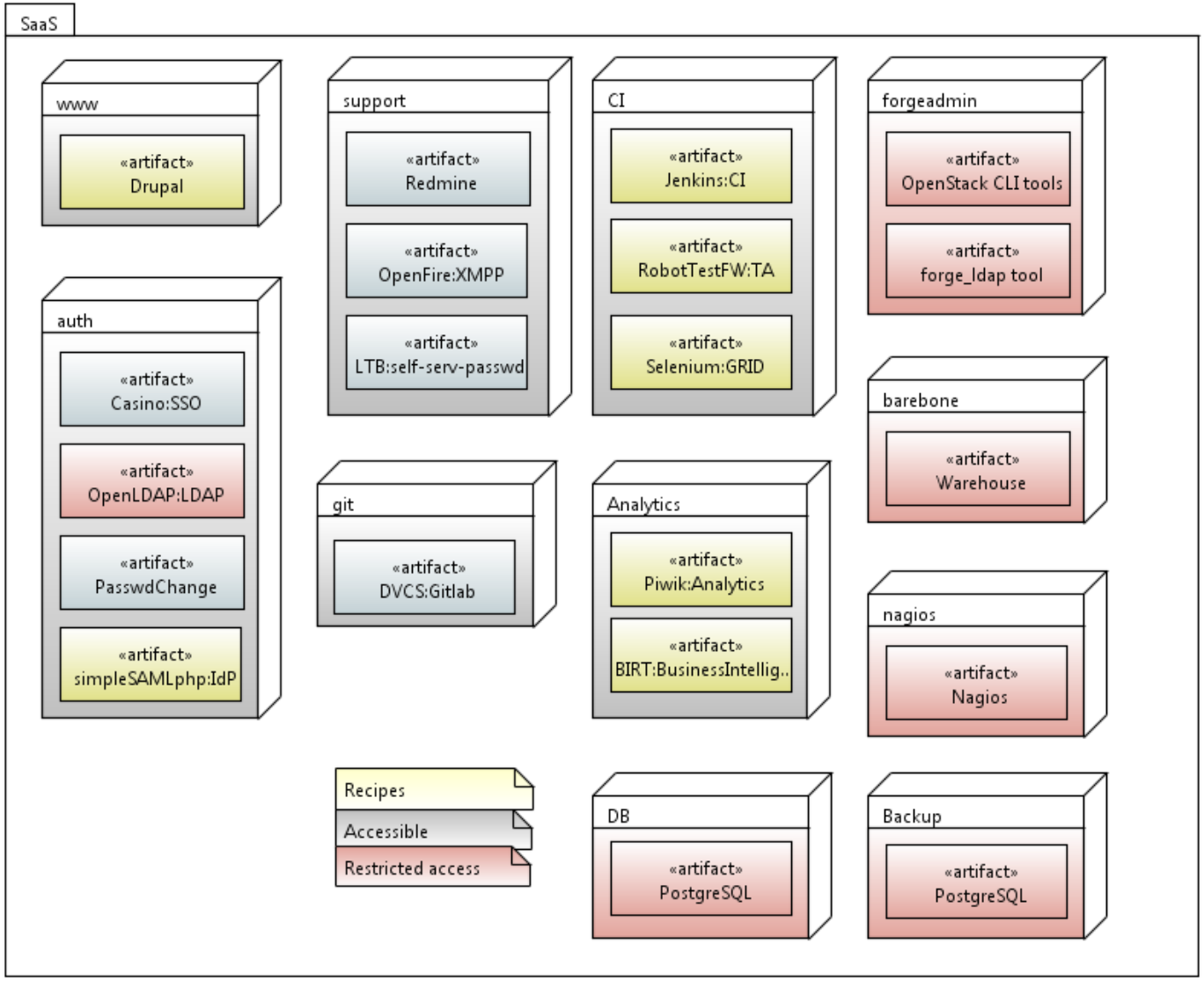
FORGE Service Lab

DIGILE

Why project management is needed in FORGE – We operate cloud infra, IaaS



Why project management is needed in FORGE – We develop services, SaaS



Why project management is needed in FORGE

Team

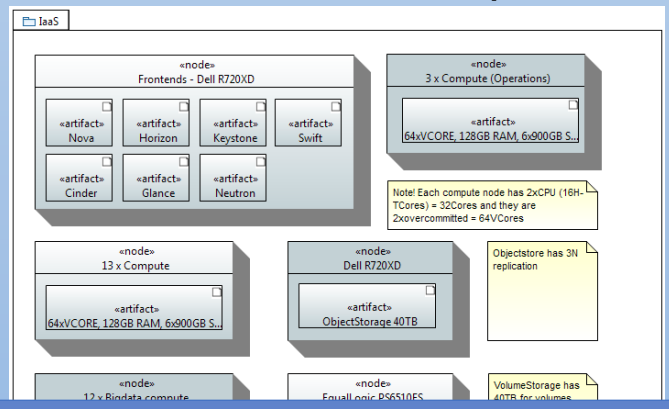
Stakeholders

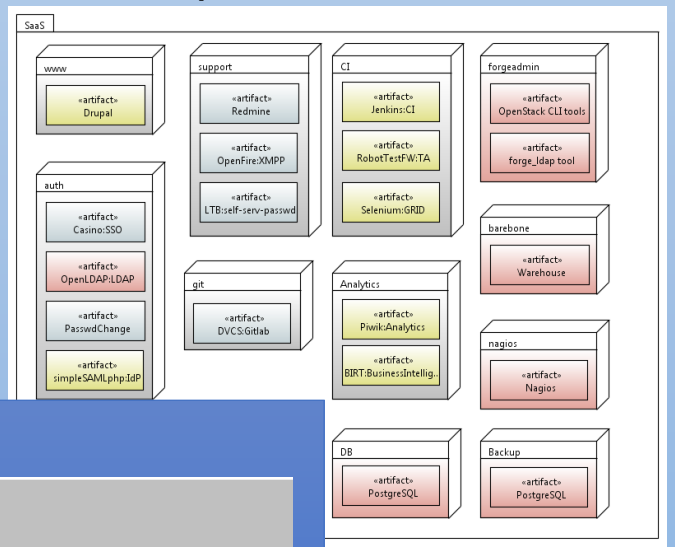
BusinessOwner

Stakeholders

Stakeholders

Operate and develop





Project facts

Project start	Sep 16 2013			
Now	May 27, 2015, 8:19 AM			
	Completed	Open	Rejected	
Days passed	618			
Weeks passed	88			
Sprints passed	44	3		Sprints remaining based on the story velocity
Stories completed	476	156	84	
Stories completed in one sprint	10			
Issues completed	2906	534	302	
Issues completed in one sprint	66			
Sprints passed	44	12		Sprints remaining based on the all issues velocity

Issue status

issue_status	issues2	story_points4	estimated_hours4
Closed	2906	2206	6716.5
In Progress	27	16	82
New	250	161	251.5
Rejected	302	165	727.5
Reopened	24	16	15
Resolved	216	84	276
Reviewed	17	45	101
Grand Total	3742	2693	8169.5

Stakeholders

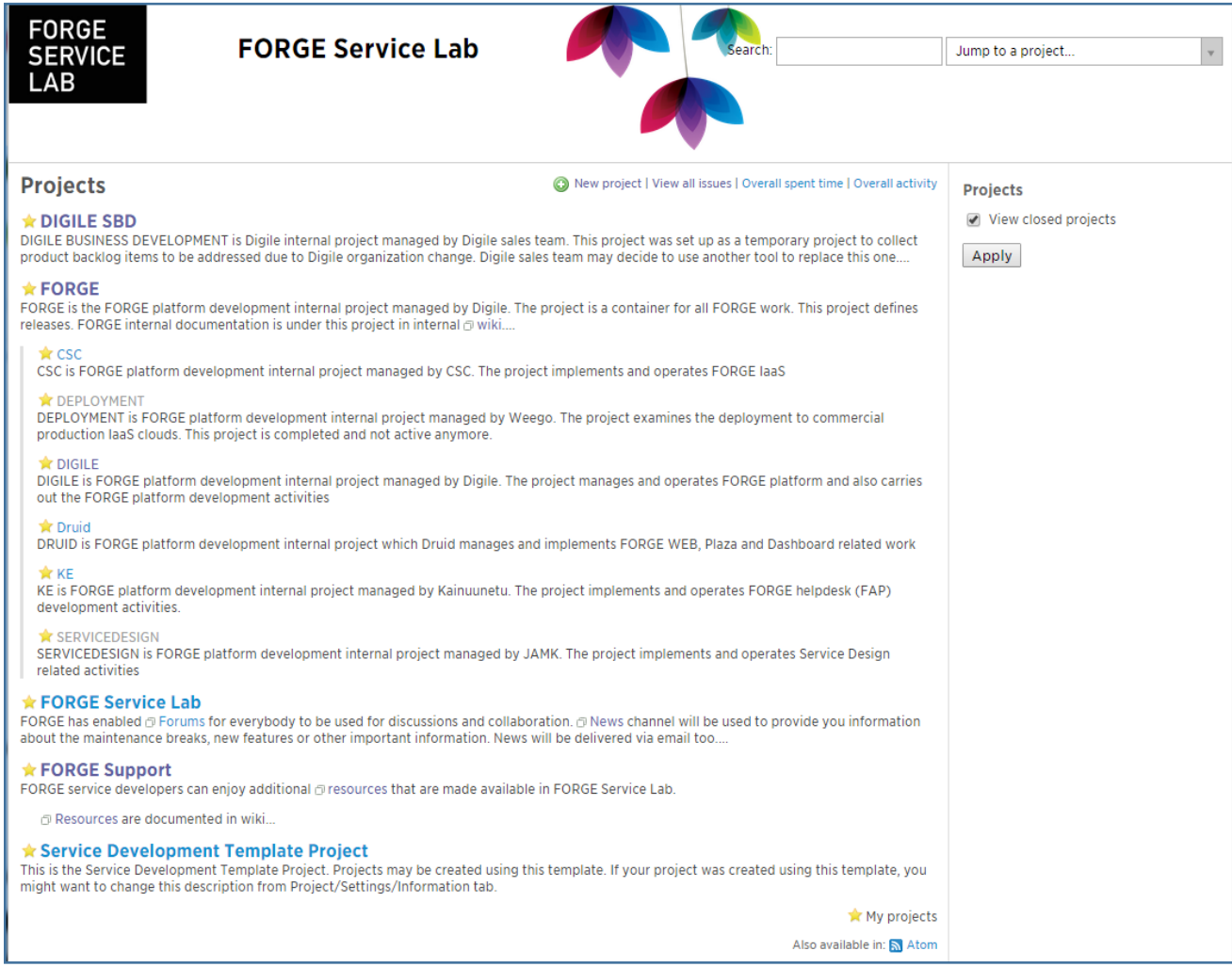
Stakeholders

FORGE project management terminology

- Definitions
  - **Project** consists of a temporary endeavor undertaken to create a unique product
  - It has an **input** and **engine** that then turns input to an **outcome**
- Scrum
  - Scrum is an **agile** method to manage projects
  - Content is represented as **User stories**. User stories are input to the projects to be implemented and completed
  - Content is stored and ordered in the **Backlogs**
  - **Storypoints** indicate the relative size of the user story compared to other stories
  - **Srint** is two week period where one project implements selected user stories from the backlog
- Scrum roles
  - **Product owner** ensures the quality of user stories in backlogs
  - **Project (team)** implements user stories of it's sprint backlog
  - **Scrum master** facilitates the scrum process
- FORGE
  - FORGE uses **scrum** methodology with incremental and multi-project flavor
  - In FORGE the **release backlog** contains common targets and content for all projects for two month period
  - Maximum ROI with minimum effort

FORGE project portfolio

- **DIGILE SBD** is SBD team's own project managed by them
- **FORGE** top level project contains common targets for all projects
  - **CSC** project implements IaaS related items
  - DEPLOYMENT project implements deployment to production experiments and examples. CLOSED
  - **DRUID** project implements web, dashboard and plaza
  - **DIGILE** project implements the digital supporting services and miscellaneous
  - **KE** project implements support function
  - SERVICEDESIGN project implements service design related items. CLOSED
- **FORGE Service Lab** if for collaboration with forums and communicating announcements for all
- **FORGE Support** contains support and documents for service developers.



Release and Sprint planning roles

Business owner

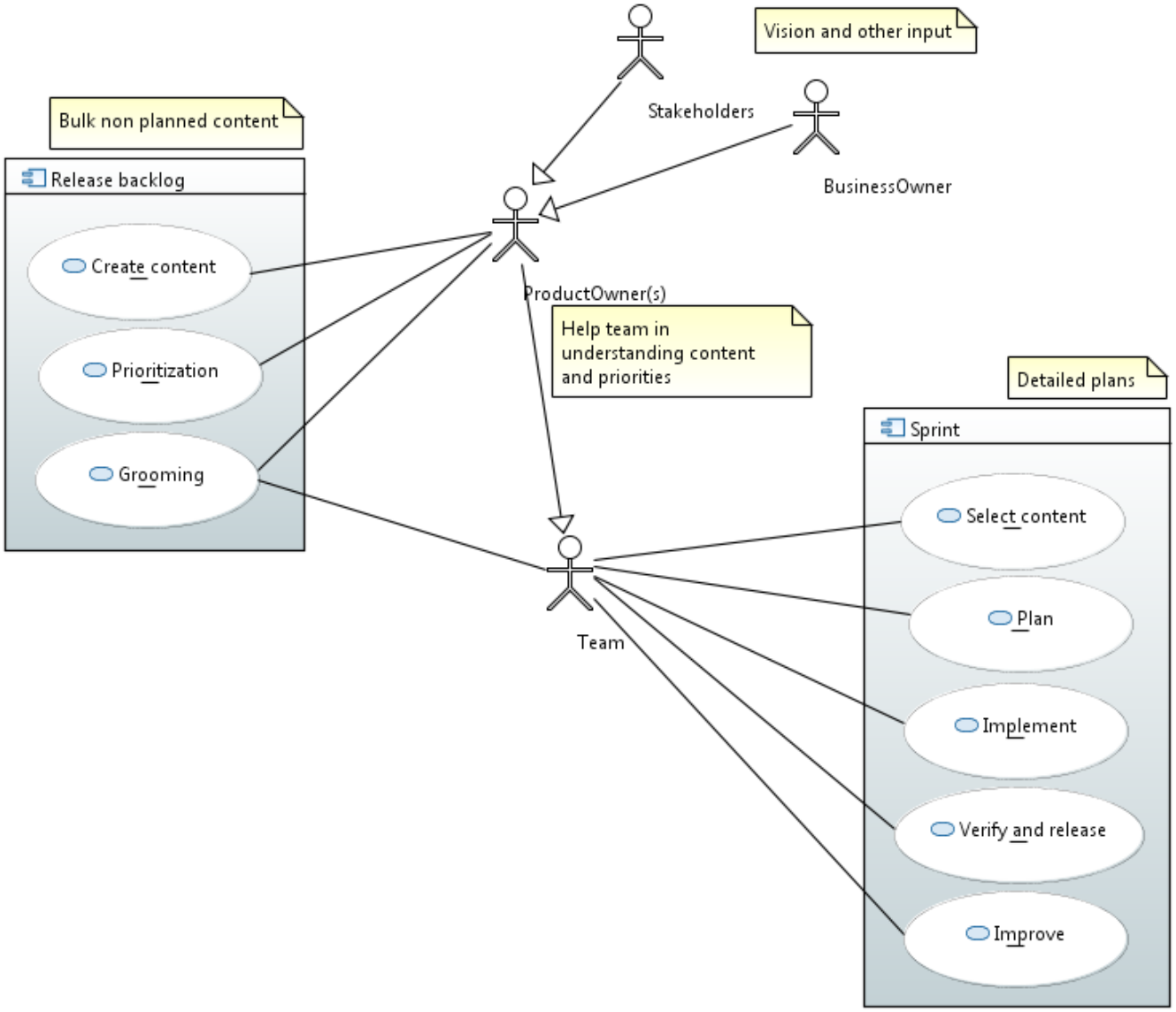
- Provide vision and other input

Product owner

- Prioritize and groom backlogs
- Help projects in planning the content
- Explain content for teams

Team and scrum master

- Select, plan and implement the content
- Followup and improve processes and practices



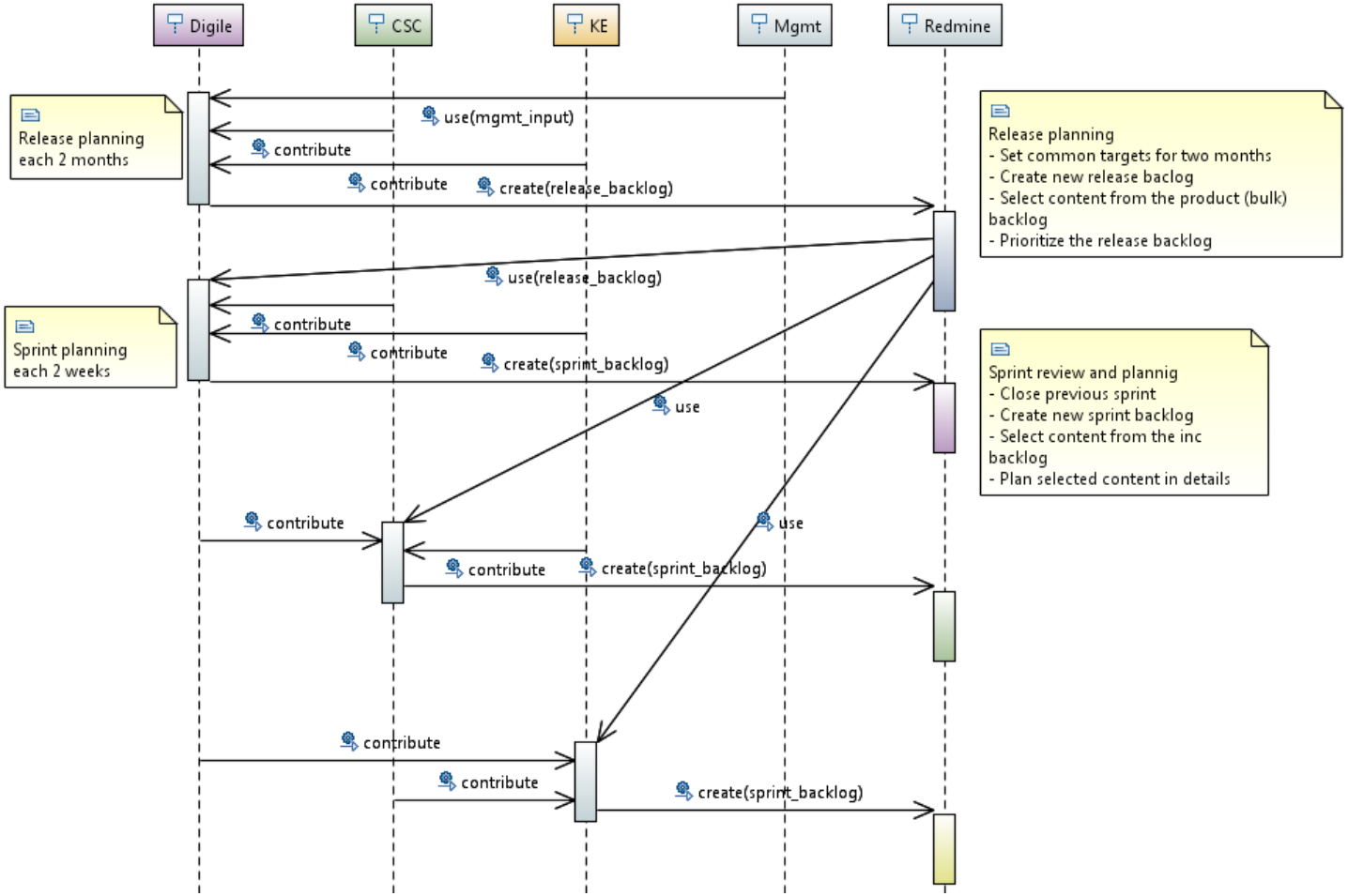
Multiproject cross-project planning

Release planning

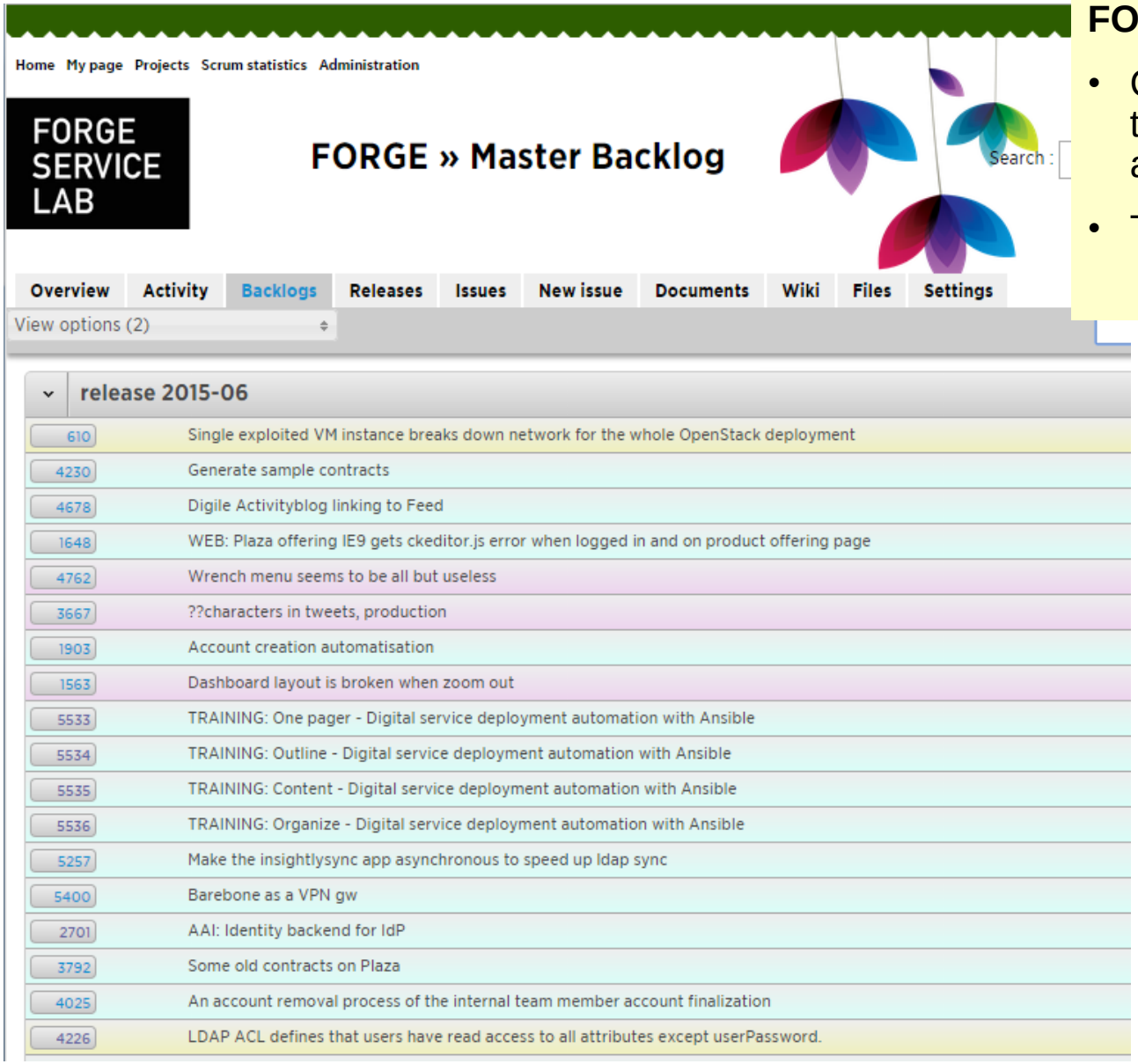
- High level program planning for two month period
- Common targets and user stories for all projects

Sprint planning(s)

- Detailed sub-project planning for two week period
- Each project selects set of user stories and defines tasks for its sprint



FORGE program release backlog



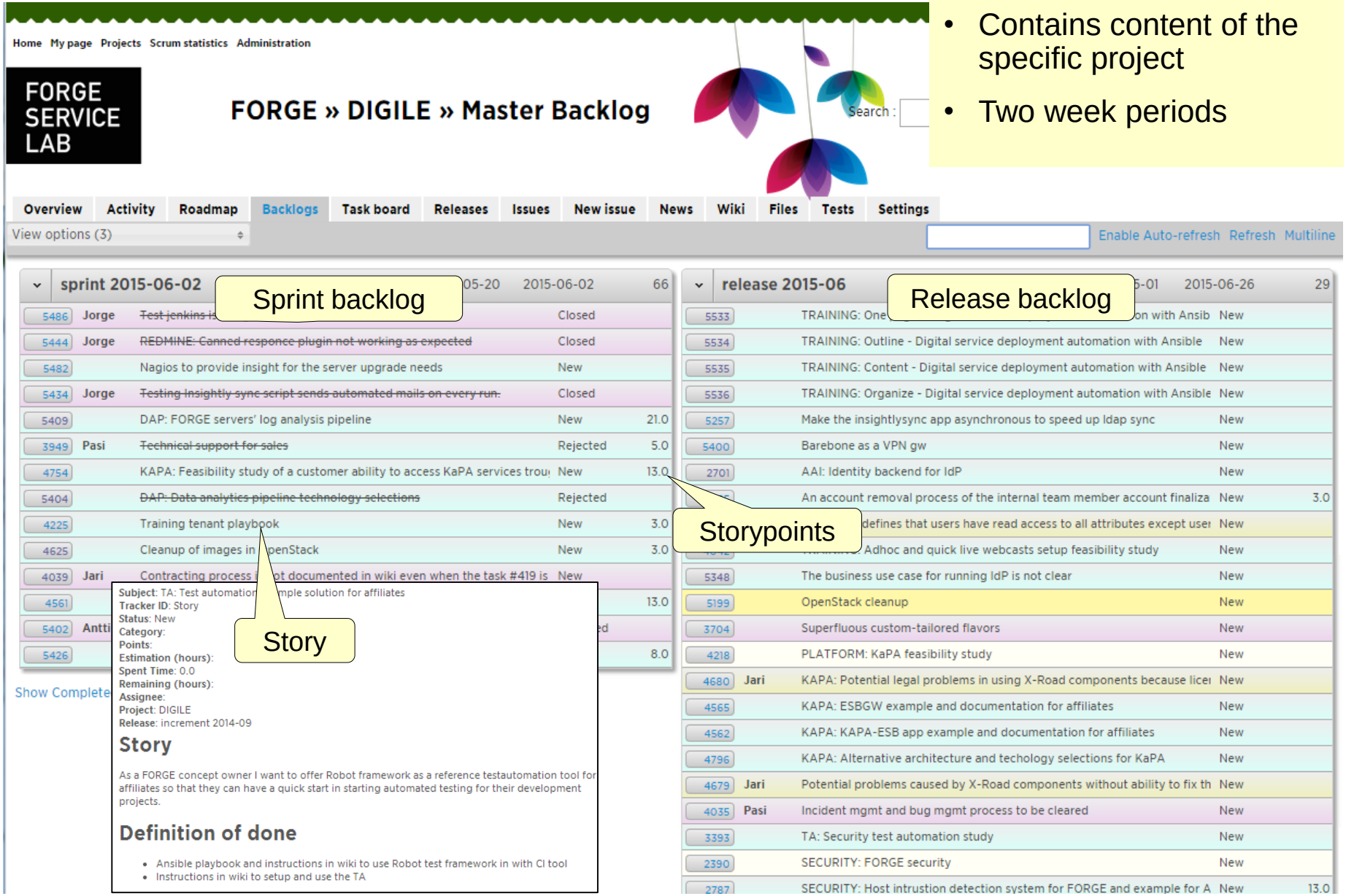
FORGE top level project

- Contains common targets and content for all sub-projects
- Two month periods

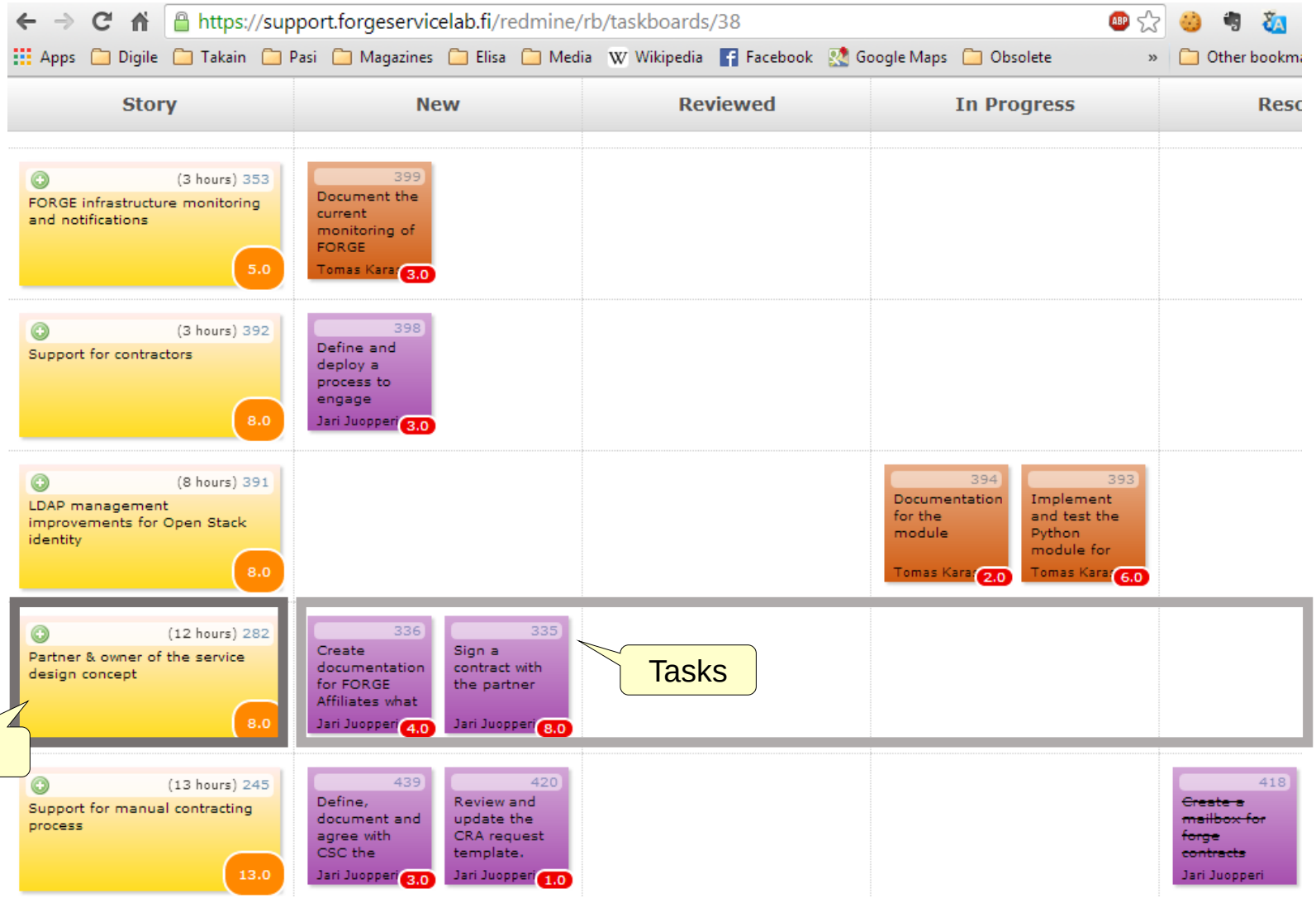
Project's sprint backlog

Sub-project

- Contains content of the specific project
- Two week periods



Project's detailed sprint plan



Program release planning procedure

Targetsetting

- Define high level targets for the release
- Define detailed targets for the release

Grooming

- Content creation and prioritization
- Move the relevant user stories from product backlog to the release backlog
- Create new user stories and split too big user stories if needed
- Prioritize the user stories

Project sprint planning procedure

Preconditions

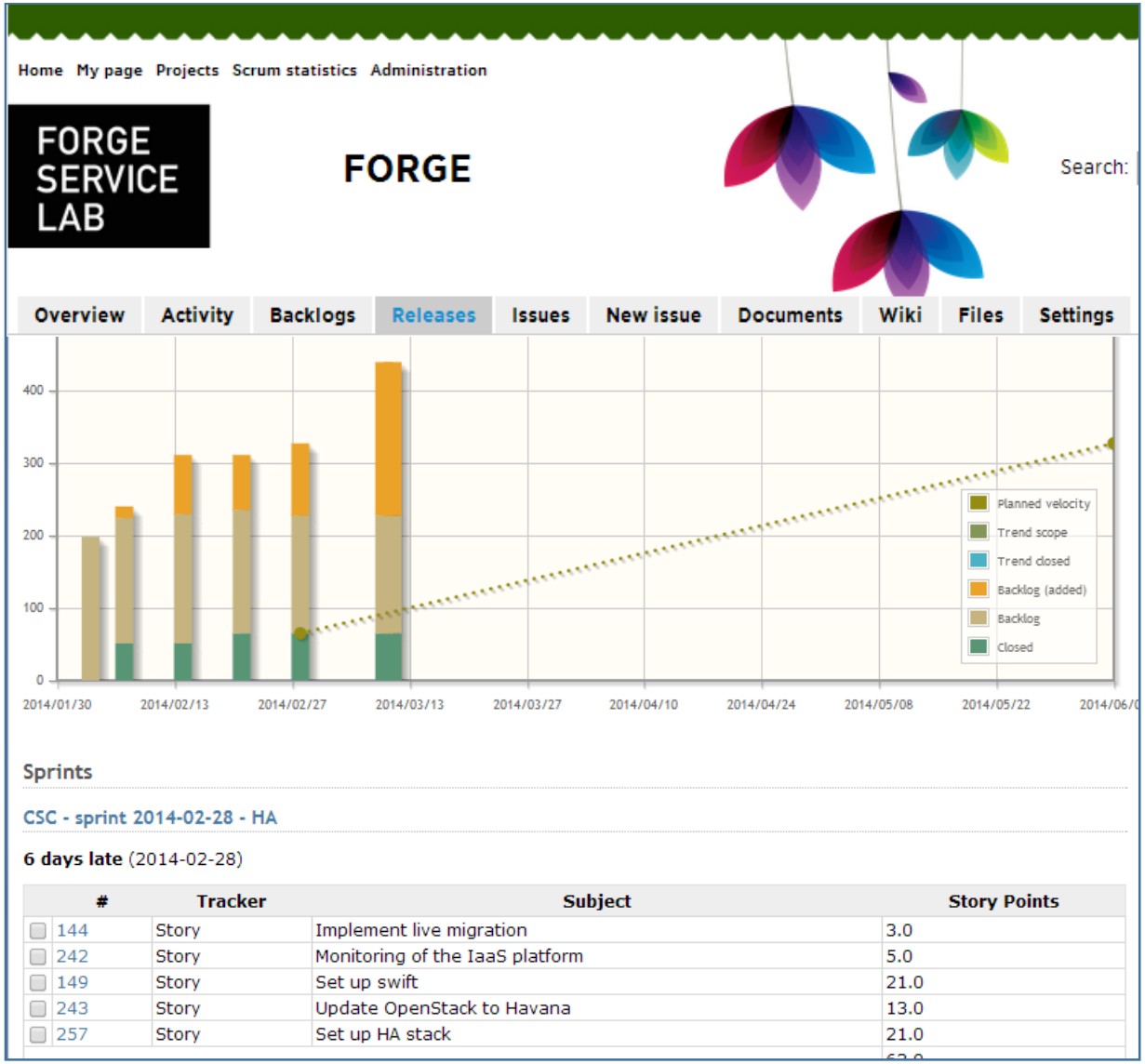
- Release planning is done and targets are available
- Previous sprint is reviewed and closed
- Retrospective held and improvements identified

Planning

- Revisit release targets
- Plan stories one by one
  - Define story points for the stories if not already done
  - Take the story from release backlog to sprint backlog
  - Product owner explains the story
  - Discuss and split the story if needed
  - Create and assign tasks according to team members volunteer to take them
- Review the developed taskboard with product owner and start sprint

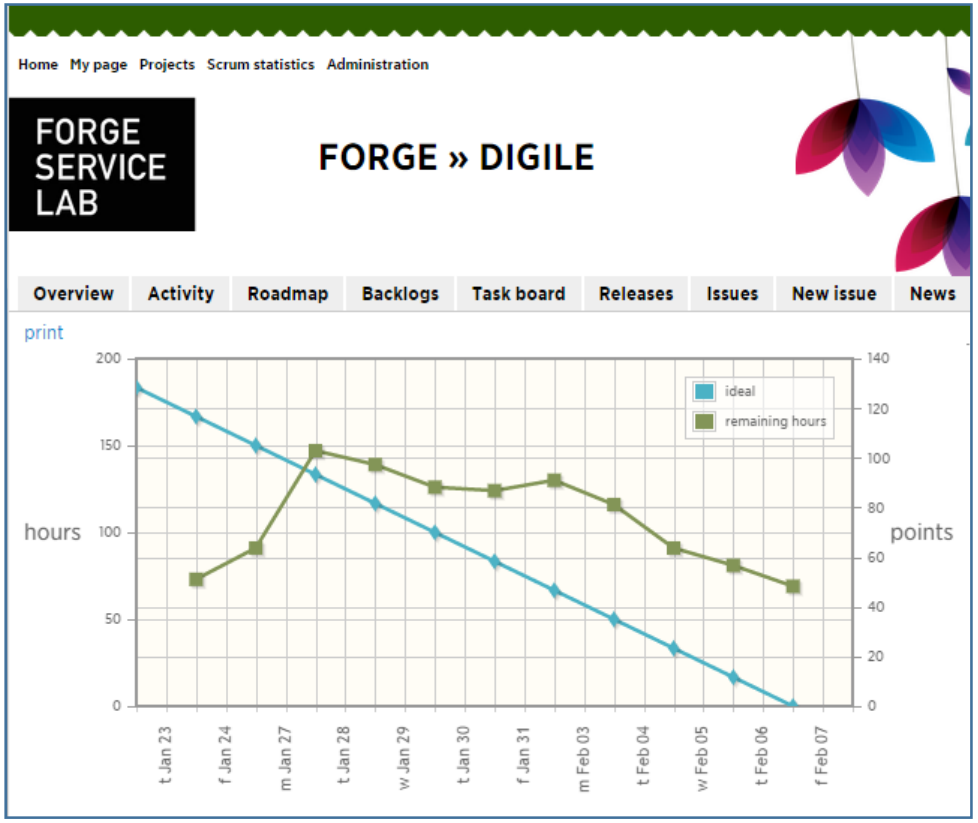
Program release metrics and release notes

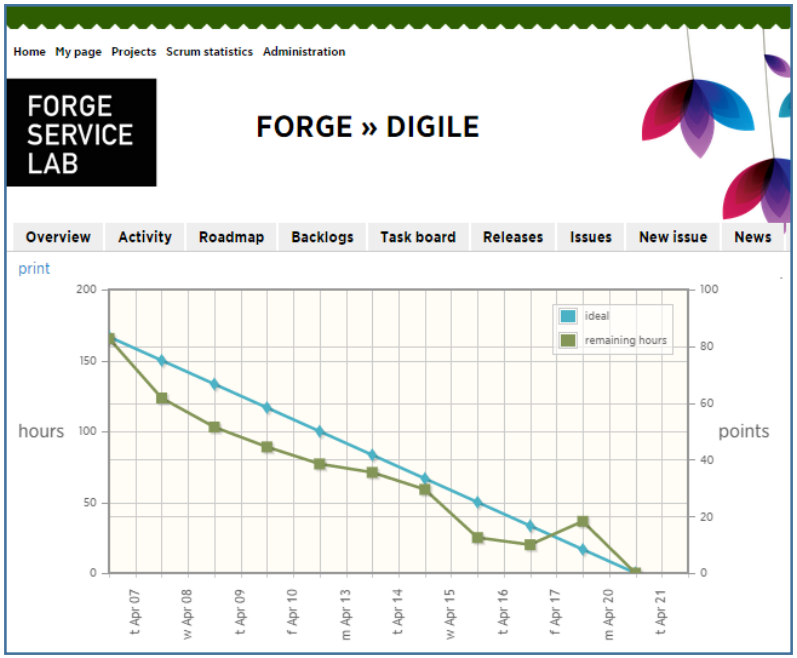
- For each release the program start to get data about completed vs. remaining content
- It's possible to estimate when the whole release can be completed
- If more content is added on the go, then then the time increases too
- The estimate is as good as the quality of storypoints given
- We have decided that we'll rather keep the deadline fixed and may change the amount of content



Project sprint burndown metrics

- Each project gets the velocity data for each of it's sprint
- Velocity tells how much content (storypoints) a project completed in the sprint
- Based on the velocity history, we can estimate how much content the project is able to complete in sprints
- All extra tasks given/taken during the sprint or other changes will impact on velocity





Program release metrics – all time