Earned Value in Agile:
The Definition of done in Agile Software development

EVA 16, London, June 14th – 15th

Kjetil Strand, Promis AS

PROMIS
Project Management • Information Systems

Outline of the talk

- Background – Earned Value Analysis in Agile
- Work Breakdown Structure in Agile
- A Project Execution Model based on PS2000 Agile
- The Control Gate following each Sprint: Definition of Done
- An Estimation Model based on the Execution Model
- A practical example: Cashing in and Monitoring Earned Value

PROMIS
Project Management • Information Systems

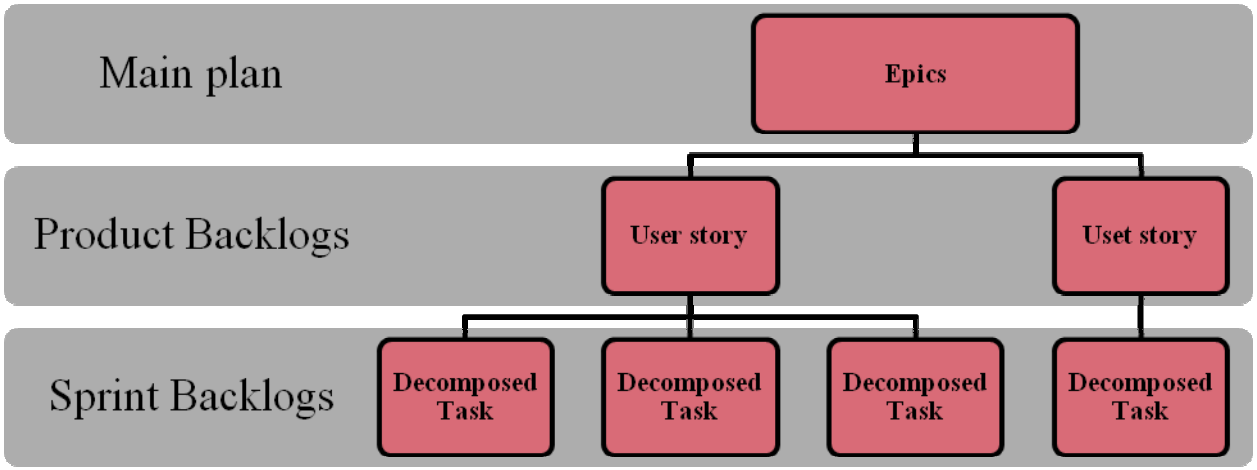
Earned Value Analysis in Agile Projects

- Some resistance against Earned Value Analysis in the agile community
- Some regard established project management knowledge areas as waterfall (you have to establish a project budget)
- This talk will demonstrate that Earned Value analysis fits well together with agile practises
- We present a framwork within which earned value could be monitored throughout the history of agile software development projects

PROMIS
Project Management • Information Systems

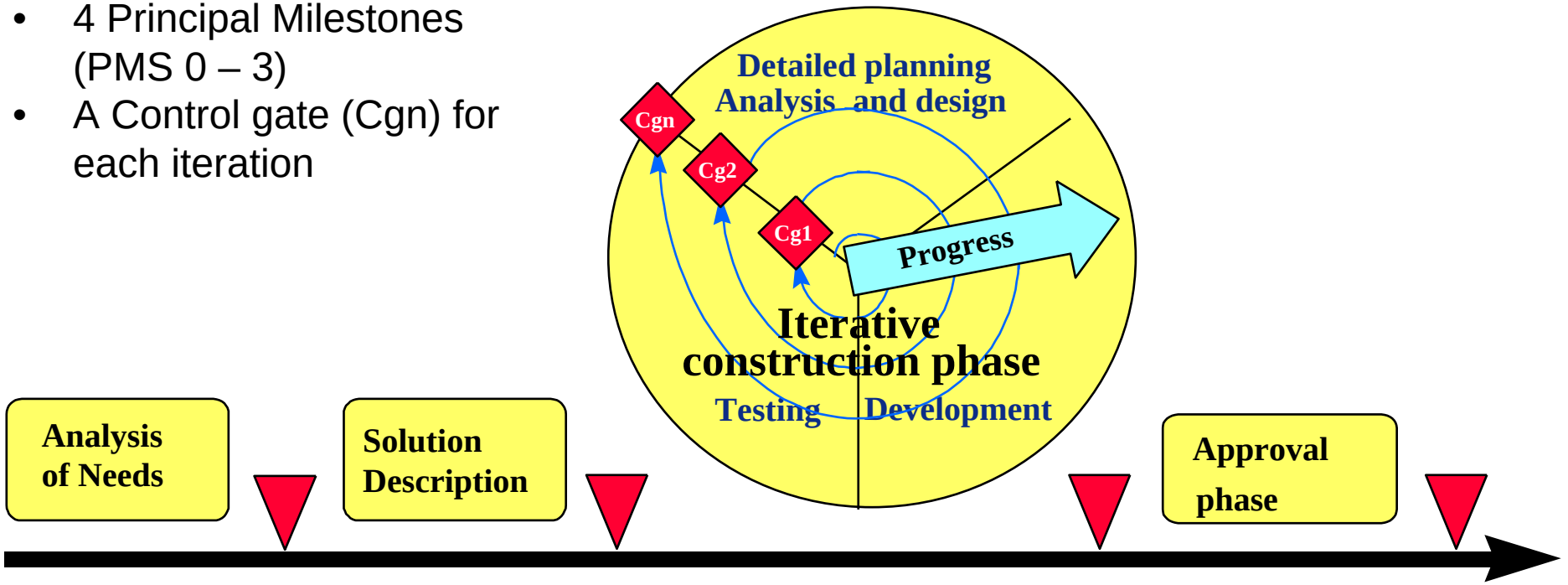
Work Breakdown Structure in Agile

- The two main levels in the project plan are epics and user stories
- The project scope is described by epics (high level user stories)
- The project budget is distributed on this epics level
- Further detailing on the user story level (product backlogs) and the sprint task levels
- Full tracability top down and bottom up on cost and progress



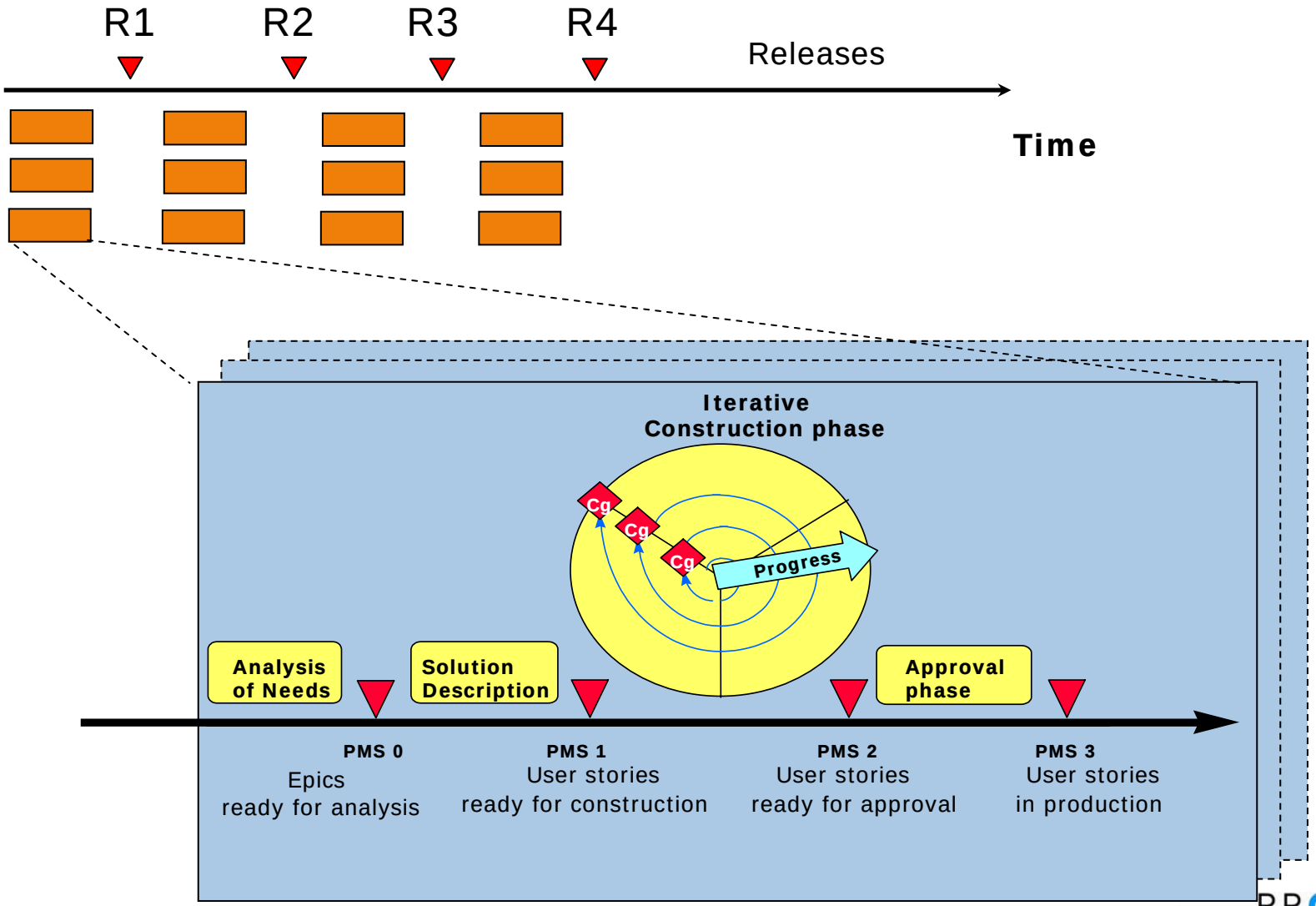
The execution model in PS2000 Agile

- 4 Principal Milestones (PMS 0 – 3)
- A Control gate (Cgn) for each iteration



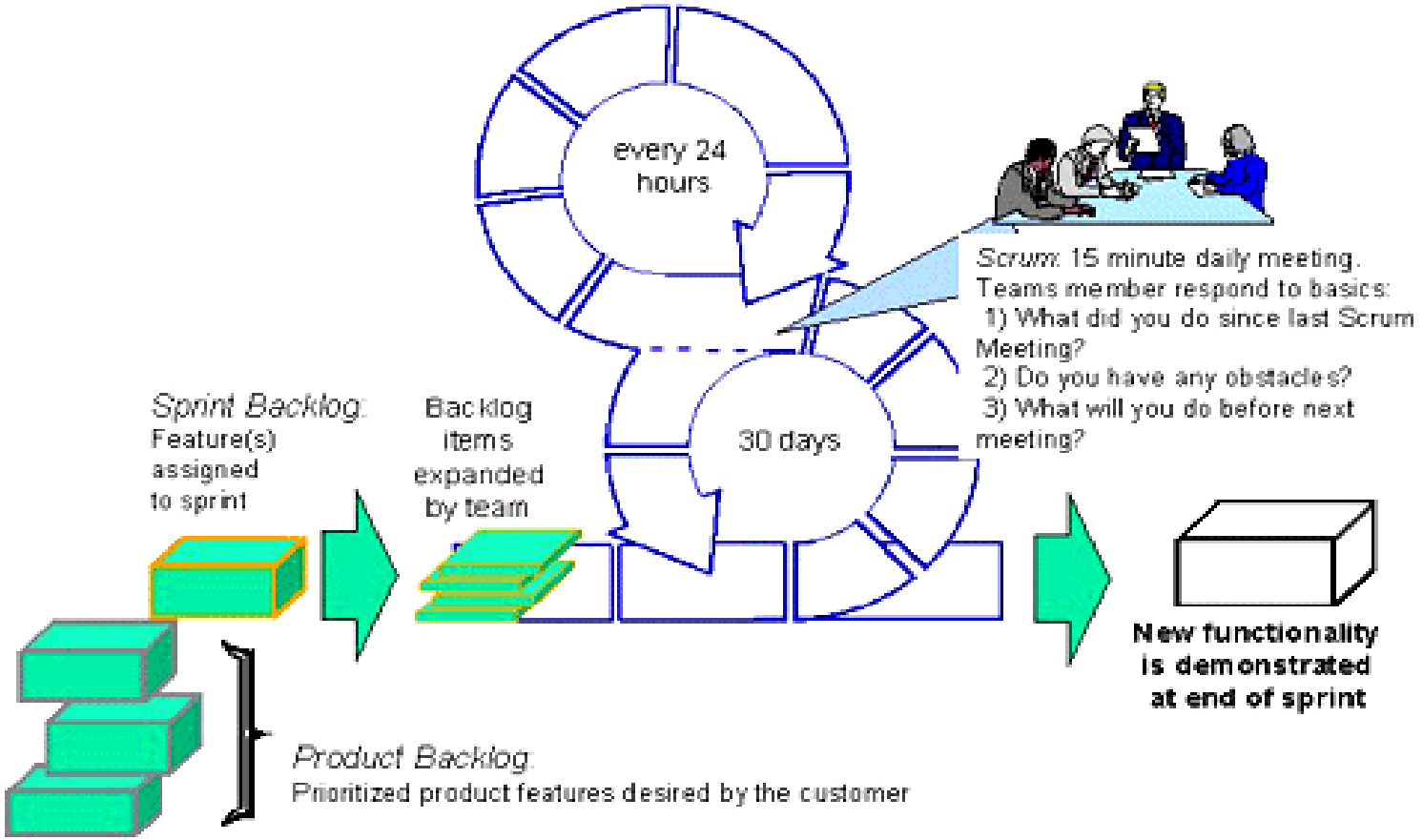
PROMIS
Project Management • Information Systems

Large projects: The Execution Model is repeated for each release



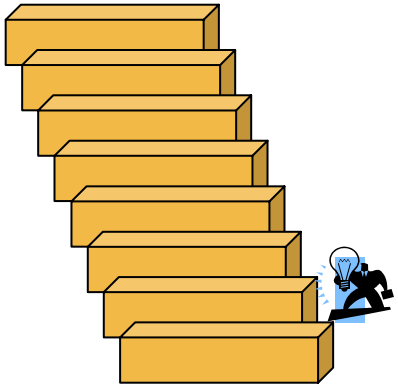
PROMIS
Project Management • Information Systems

SCRUM: Each sprint is an iteration



PROMIS
Project Management • Information Systems

The Anatomy of the Sprint in PS2000 Agile



Prioritized Product Backlog

-What did you do?

-What will you do?

-Any obstacles?

24 h

Development

Detailed planning, analysis and design

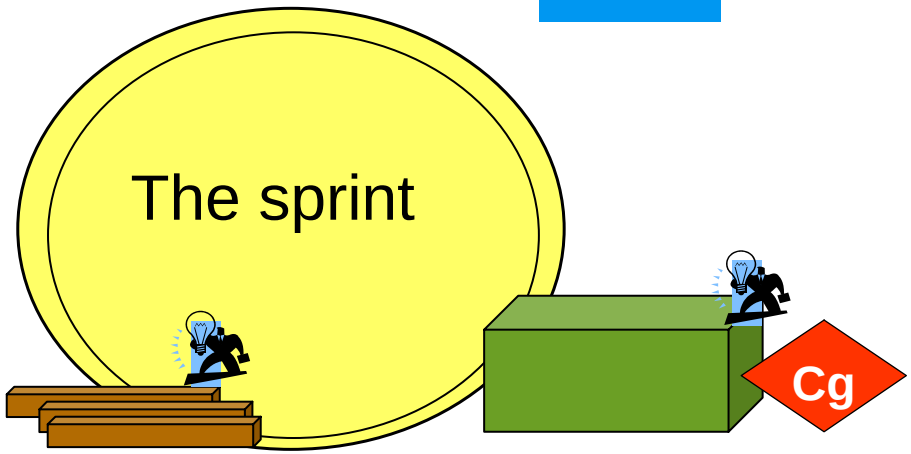
Testing

Sprint Backlog

Cg

Planning	Decomposed by the team	The delivery in the Control gate – running software and definition of done

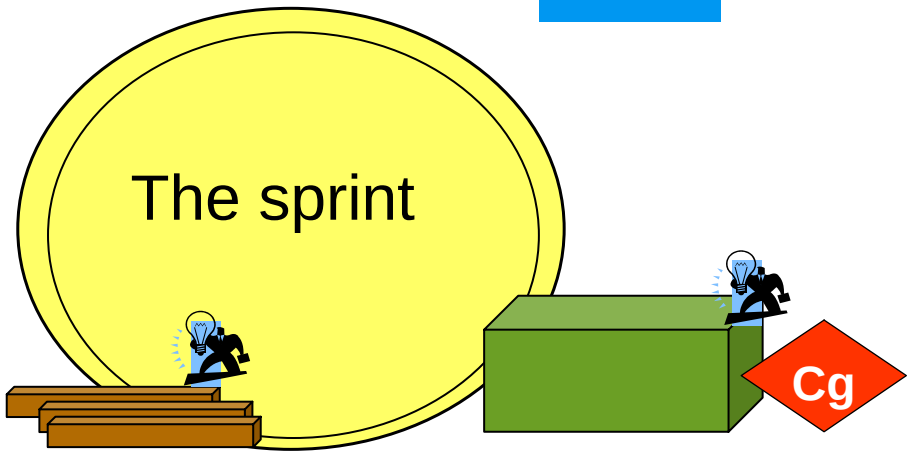
The Control gate



- By the end of the sprint, the teams demonstrate running software to the product owner(s)
- Furthermore, to check if a user story meets Definition of done, it must pass a Control gate
- The Control gate meeting is usually executed 2-4 working days after sprint demo (by this time, the teams have already executed sprint planning for the next sprint)
- In the Control gate process and the Control gate meeting a lot of representatives from the Customer side are participating: Product Owners, Test, Architecture, Operations, and project management
- In the Control gate meeting the Customer gives feedback on all parameters of ‘Done’ to the Vendor

PROMIS
Project Management • Information Systems

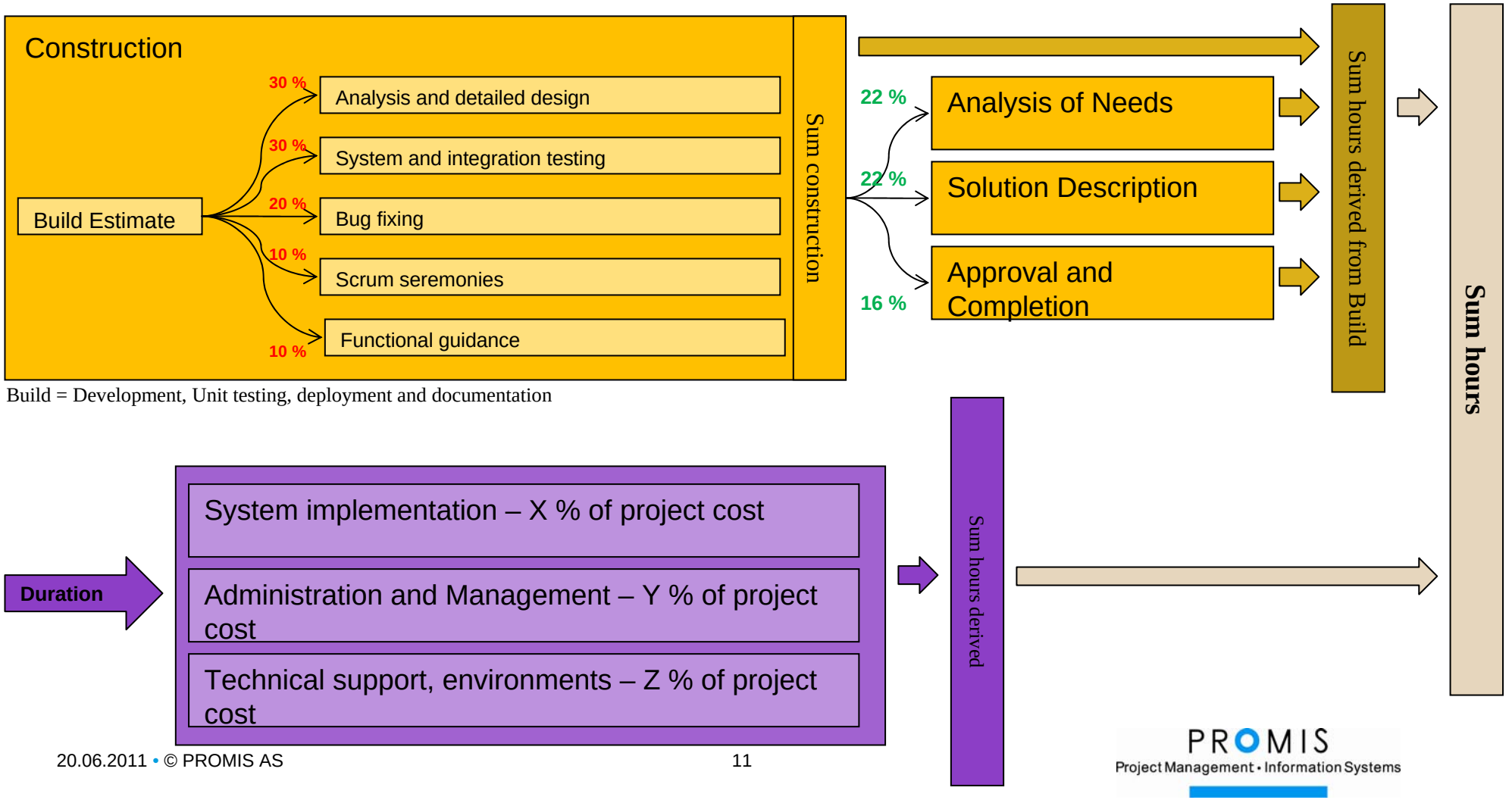
Definition of Done



- The user stories are verified on a stable test environment
- Do the user stories meet the acceptance criteria?
- Is the software well documented (user documentation, system documentation, installation and operations documentation)?
- Are the tests documented?
- Is the code of good quality?
- Are other architectural constraints and guidelines met?
- All these requirements should be fulfilled to meet the definition of done
- The control gate meeting itselt may handle a number of delivered user stories in a relatively short time (e.g., 30 user stories in 15 minutes)

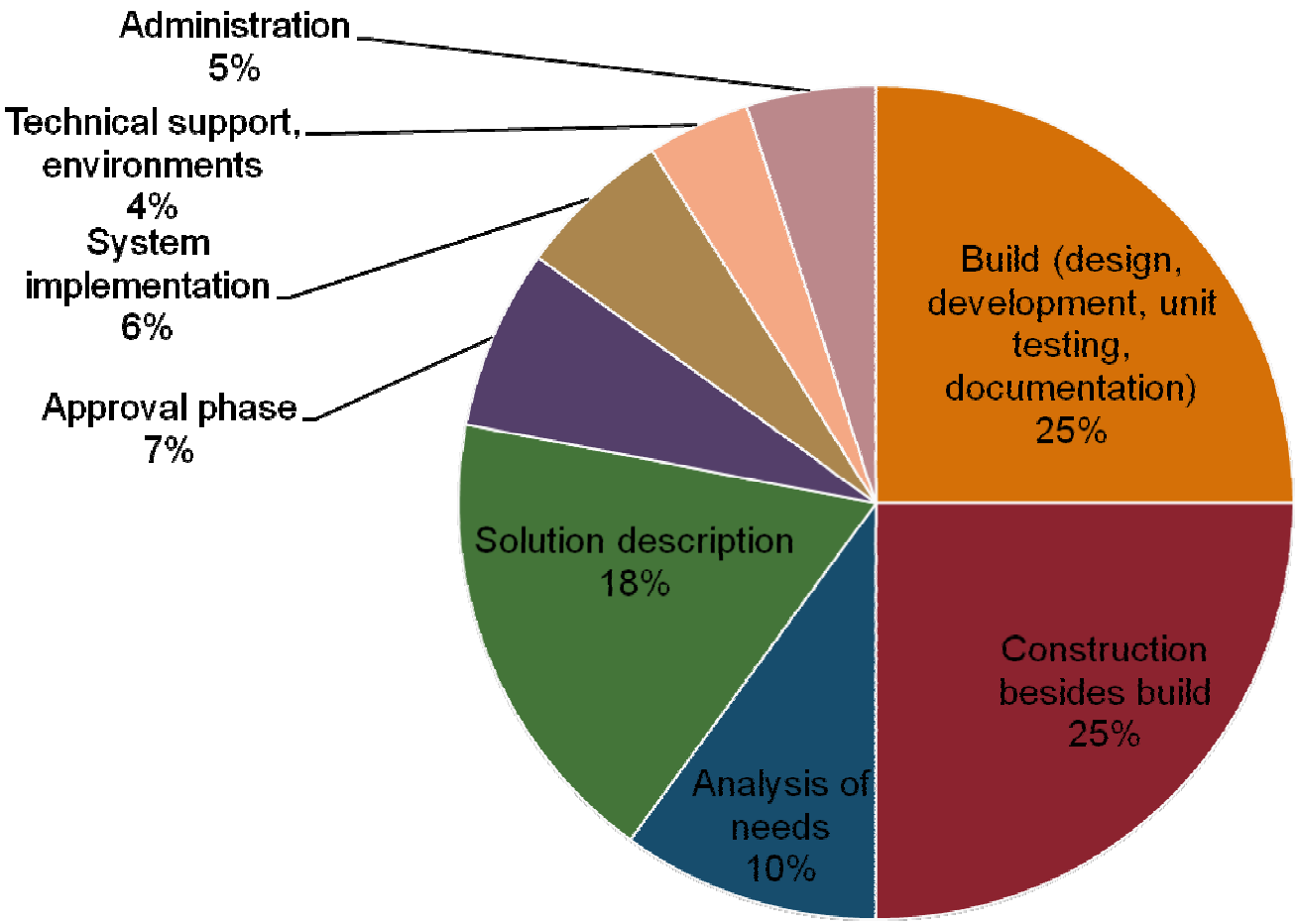
PROMIS
Project Management • Information Systems

An Estmation Model with Build Estimate as the Main Driver



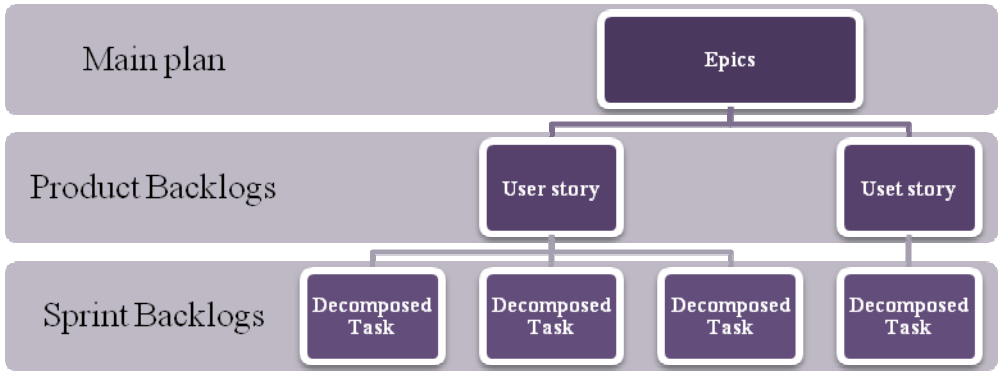
PROMIS
Project Management • Information Systems

An implementation of the Estimation Model

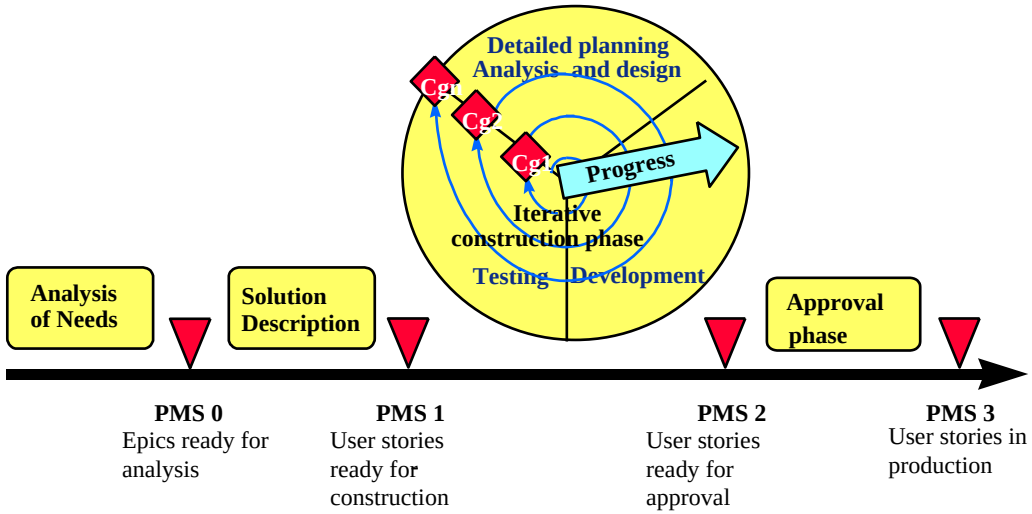


Mapping the estimation model to earned value in the principal milestones

Milestone	Earned value
Epics not started on	0%
Epics ready for solution description (PMS 0)	11 %
User stories ready for construction (PMS 1)	31 %
User stories ready for approval (PMS 2)	86 %
User stories in production (PMS 3)	100%



- Earned value in each principal milestone is computed according to the estimation model
- Nothing else than progress on epics and user stories count as earned value
- Other activities in the project are considered useful only to the degree that they support progress on epics and user stories



PROMIS
Project Management • Information Systems

Control gates verify that user stories are ‘Done’

- When approved of in the control gate, we may cash in 86% of the budgeted value of the user story (according to this implementation of the estimation model)
- User stories not approved of, are not cashed in
- These user stories remain on EV = 31% of budgeted value, together with other user stories still in construction
- These user stories are returned to the product backlog and prioritized for the ongoing or future sprints
- Most commonly, the team will commit to deliver these user stories in the ongoing sprint, in addition to the commitment from their sprint planning
- When passing the control gate, only the approval phase and system implementation remain – these activities are estimated to 14% of project cost (according to this implementation of the estimation model)

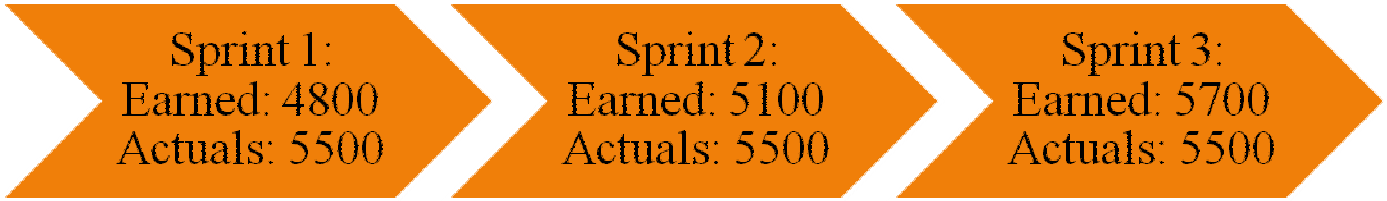
PROMIS
Project Management • Information Systems

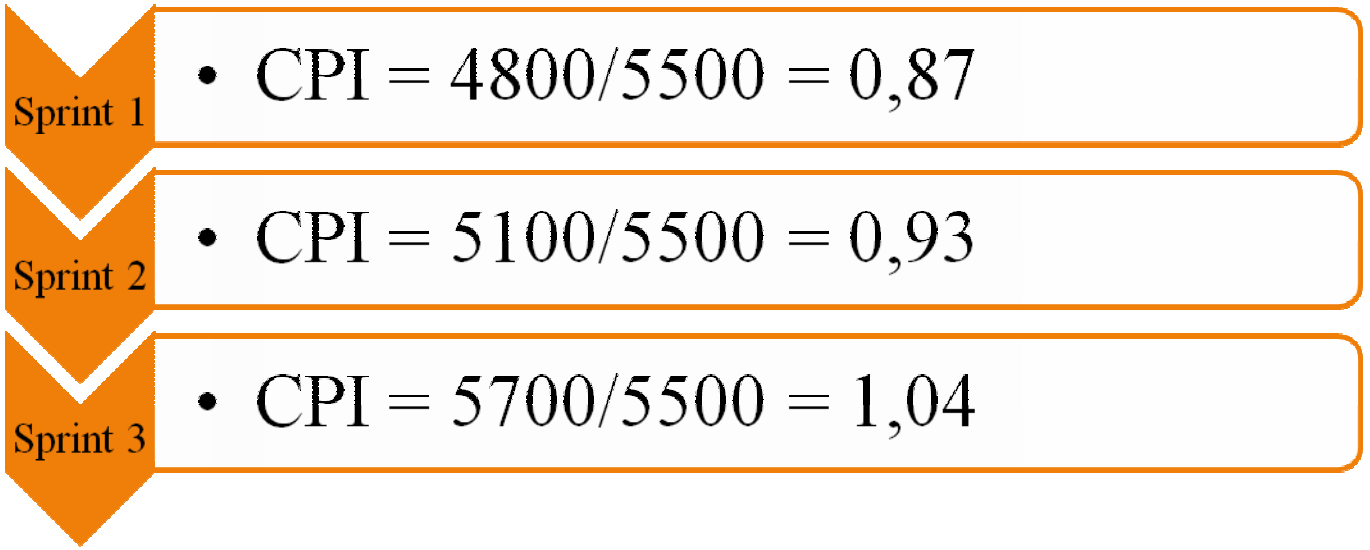
The framework is applied in a large system development project

- A project in the Norwegian public sector
- Duration 2008 – 2012, worth more than 100 MILL €
- 3 vendors, 13 parallell sprint teams
- The execution model in this project is based on the PS2000 agile contracting standard
- The framework has been a partly success, but with some challenges

PROMIS
Project Management • Information Systems

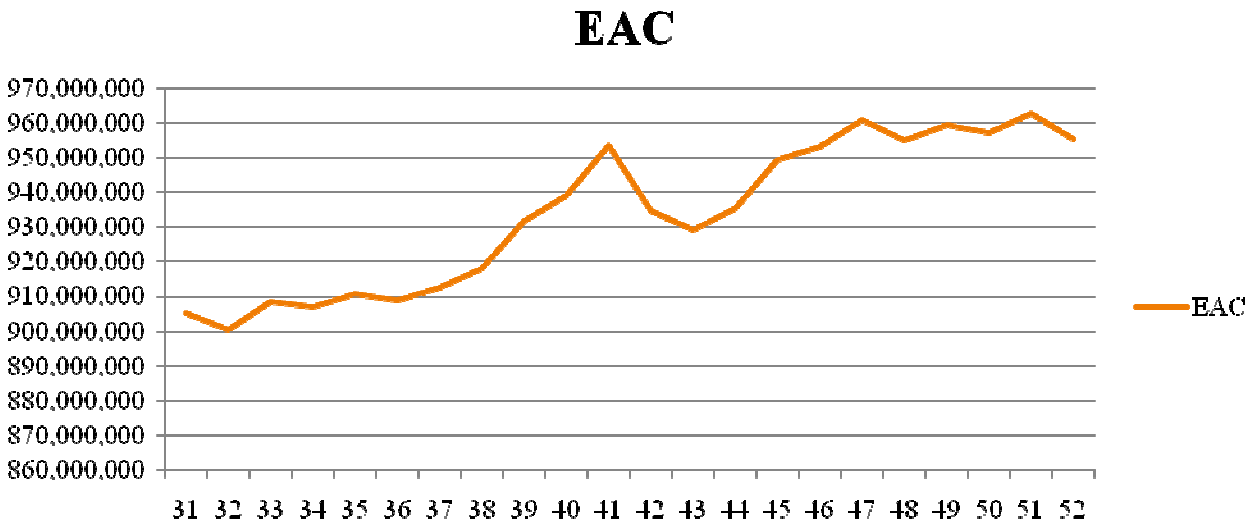
Local CPI tends to vary from sprint to sprint

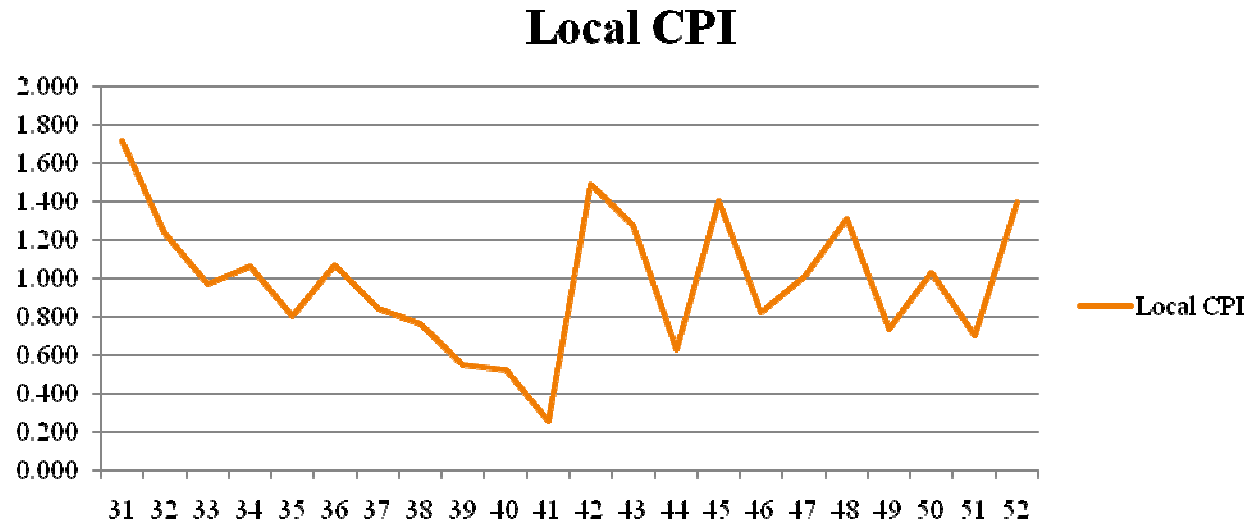




The last 22 sprints in the aforementioned project

Date	Sprint	Local CPI	Accum-ulated CPI	EAC
14.01.2010	31	1,717	1,072	905 239 418
28.01.2010	32	1,239	1,081	900 386 434
18.02.2010	33	0,971	1,075	908 533 357
10.03.2010	34	1,059	1,074	907 062 519
13.04.2010	35	0,803	1,060	910 828 360
06.05.2010	36	1,066	1,060	908 928 974
27.05.2010	37	0,843	1,051	912 736 258
17.06.2010	38	0,762	1,040	918 163 248
10.08.2010	39	0,545	1,009	931 881 303
30.08.2010	40	0,520	0,994	939 354 719
20.09.2010	41	0,254	0,967	953 541 642
14.10.2010	42	1,491	0,992	934 825 418
09.11.2010	43	1,275	1,001	929 130 556
30.11.2010	44	0,625	0,990	935 510 853
21.12.2010	45	1,401	1,002	949 578 123
13.01.2011	46	0,823	0,997	953 357 009
02.02.2011	47	1,008	0,998	961 012 906
02.03.2011	48	1,307	1,006	955 300 378
17.03.2011	49	0,732	0,999	959 641 416
08.04.2011	50	1,029	1,000	957 193 635
06.05.2011	51	0,700	0,992	962 920 388
26.05.2011	52	1,399	1,002	955 401 148





Pros & Cons of the presented framework

On the one hand…

- During sprints 37 - 41 from the previous slide, the local CPI deteriorated considerably
- This was mainly due to a prolonged approval phase of the largest release in the project
- Large variations in local CPI (and because of this, in the EAC), may be hard to communicate to the steering committee

On the other…

- The framework is easy to implement and maintain
- No special tools are needed:
  - Budgeted hours on epics and user stories
  - Issue tracking system like Jira to provide the status of epics and user stories after each sprint
  - A time tracking system to gather the actual worked project hours on a weekly basis
- Robust enough to provide the information needed on project progress

PROMIS
Project Management • Information Systems