



# Tools for Software Project Management

- 2-plan
- Aceproject
- Apache Bloodhound
- Assembla
- Basecamp
- Bug-Genie
- Clarizen
- Collabtive
- Feng Office
- GanttProject
- Gemini Tracker
- MS Project (covered by LU license)
- LibrePlan
- OpenERP
- Project Open
- TACTIC
- Teamwork
- Trello
- ....

Pick 3 to evaluate!

## Your Case: DauMob Ltd



Fictive, but realistic large-scale software dev company

### Case-based teaching / learning

- An active learning strategy based on complex, real-life scenarios
- Stimulates analytical thinking skills and decision-making

## COURSE SCHEDULE – updated weekly!

## Course Info – [cs.lth.se/etsf01](https://cs.lth.se/etsf01)

Project Sign-up

Course Literature

Lectures

Exercises

Exam

### Project description

## ETSFo1 - Software Engineering Process - Economy and Quality

Välkomna till upprop och första föreläsning må 20 mars, kl 13.15 i M:A (OBS! M-huset)

Att göra nu:

- Anmäl dej till projektgrupp och övningsstämman [här](#) (om du inte redan gjort så)
- Bekanta dej med projektet [här](#)
- Beställ kursboken nu, t.e [här](#)

Log

170310 Projektanmälan öppnad [här](#). [Projektbeskrivning](#) publicerad. Kursschema uppdaterat.

170306 Kursprogram & schema publicerat, se panelen till höger. OBS! Schemat innehåller [läsinstrux](#) o kommer att uppdateras löpande - [Här](#)

Sidansvarig: [Elizabeth Blomqvist](#) | 2017-03-10 [Logga Ut](#)

### Course program

Course schedule w läshänvisningar

### FACTS ABOUT THE COURSE

Credits: 4 hp

Study period: VT2, 2016

Schedule (TimeEdit): [Click here!](#)

Course responsible: [Elizabeth Blomqvist](#)

Student representatives: C - TBD

Course plan: [Here](#)

Course program: [Here](#)

Course schedule: [Here](#)

Literature: see [course literature page](#)

### GENERAL LINKS:

Rules regarding compulsory course moments (available in Swedish only): see [here](#).

Cooperation or Plagiarism: Rules can be found [here](#).

Course evaluation (available in Swedish only): see [here](#).

Läsvecka		Lectures <small>Red references are final, Blue are preliminary</small>	Exercises <sup>1</sup>	Exercise topic	Project deliveries <small>Ex2-3 in moodle, final via email</small>
Mar 20	1	L1 Overview, Course project overview, Activity planning <a href="#">Ch 1</a> [not 1.8, 1.9, 1.11, 1.14], <a href="#">Ch 3</a> , <a href="#">Ch 6</a> [not 6.16], <a href="#">P1</a>	Exercise 1	PROJECT KICK-OFF	Define groups
Mar 27	2	L2 Course project details, Effort estimation, Resource allocation, Organisation <a href="#">Ch 5</a> [not 5.11-5.12], <a href="#">Ch 8</a>			
Apr 3	3	L3 Risk management, Agile project management <a href="#">Ch 2.6</a> , <a href="#">Ch 7</a> [not 7.3, 7.8 & 7.11], <a href="#">Ch 4.10-11</a> , <a href="#">4.13-15</a> , <a href="#">P2</a> , <a href="#">P3</a>	Exercise 2	SPA I (Student Peer Assessment)	Draft 1: 50 h prior to exercise class SPA I reports: prior to exercise class
Apr 24	5	L4 Monitor & control, SW Process Improvement, Software quality management, <a href="#">Ch 9</a> [not 9.6], <a href="#">12.4</a> , <a href="#">P4</a> [Sect 3.2], <a href="#">Ch 13</a> , <a href="#">P5</a> [Sect 1-3]			
	TUE L5	Managing people, <a href="#">Ch 11.1-3</a> , <a href="#">11.6</a> , <a href="#">11.8</a> Guest lecture: Magnus Lidholm, Sony Mobile			
May 1	6		Exercise 3	SPA II	Draft 2: 50 h prior to exercise class SPA II reports: prior to exercise class
May 8	7	L6 Portfolio & Program management, <a href="#">Ch 2</a> [not 2.9, 2.10-13], <a href="#">Ch 4.2</a> , <a href="#">Ch 10:1-2</a> , Guest lecturer: Thomas Ohlsson, SICS			
May 15	8	L7 Exam walk-through & tips.	Exercise 4	PROJECT CONFERENCE	Presentation material: 26 h prior to exercise class Final report:
		EXAM –Wed 31/5, kl 14-19, MA9			

## Examination

- Written exam based on the book, articles (P1-P5) and lectures
- Max 60 p
- Structure
  - 1 terminology: definition & examples
  - 1 practical
  - 2 essay Qs with keywords
- Project: IG / G + up to 10 BPs

Exam + bonus points	Final grade
$\geq 30$ , at least 27 for exam	3
$\geq 41$	4
$\geq 51$	5

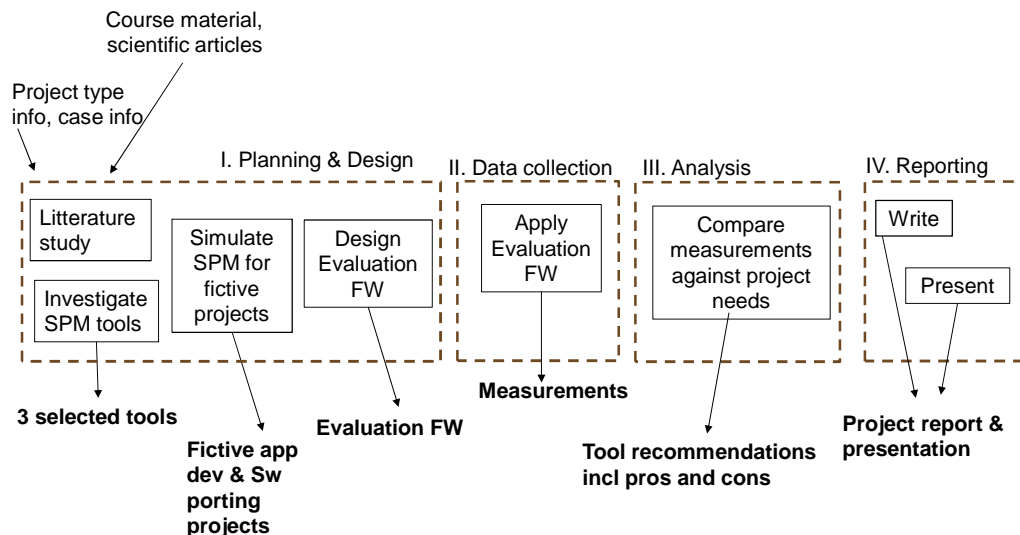
## Project assignment

**Draft 1** due on Mon 3 April

- Full outline w headings and bullets (level 1 and 2)
- Drafted content for
  - Introduction: opening, tools, case projects
  - Method: planned approach
  - Evaluation Framework: Activity planning, Effort estimation



## Course Project Activities (or Process)



## Course project activities

- Design an evaluation framework for tools
- Assess / evaluate 3 different tools
- For each tool and SPM area
  - Identify benefits and weaknesses
  - recommend suitable tool improvements
- Recommend SPM tool for each of 2 SW project types
- Report your work and its outcome
  - written report
  - oral presentation

Your choice!

Exercise 1:  
Presentation  
techniques + metrics

## Selecting tools for evaluation

Select 3 tools intended for SPM

Consider

- Access
- Available documentation
- Sufficient SPM support for case projects

## Report Structure

Max 7 pages\* in IEEE format

\* + 3 pages appendix  
for additional details

- Abstract
- Introduction: SPM, Tools, scientific references
- Method: How evaluation was performed
- Case Projects
- Evaluation Framework
- Tool Evaluation incl improvements
- Tool Recommendations per SW project type
- Conclusion
- References

## Reference credibility

Different fora has different credibility

- Scientific fora
  - Journals
  - Conferences
  - Workshops
- Non-scientific fora:
  - Textbooks
  - Journalist material
  - White papers
  - Web pages (inkl Wikipedia!)
  - ...

*Peer reviewed*

To find scientific papers

- <http://www.lub.lu.se>
- <http://scholar.google.com>
- Detective work
  - Search broad
  - Select
  - Search deep
- Search
  - Key words
  - Authors
  - References
  - Fora

} Iterate

## Example: Scientific references

- B.W. Boehm, Software Risk Management: Principles and Practices, **IEEE Software**, Jan 1991, pp. 32-41
- T. Dybå, An Empirical Investigation of the Key Factors for Success in Software Process Improvement, **IEEE Transactions on Software Engineering**, 31(5), May 2005, pp. 410-424.
- E. Bjarnason, K. Wnuk, B. Regnell. Are you biting off more than you can chew? A case study on causes and effects of overscoping in large-scale software engineering, **Information and Software Technology**, 54(10), Oct 2012, pp. 1107-1124.
- E. Bjarnason, K. Wnuk, B. Regnell. Requirements are slipping through the gaps – A case study on causes & effects of communication gaps in large-scale software development. **Proc. Of 19th IEEE International Requirements Engineering Conference**, pp. 37-46, 2011.

# Method section – Research approach

## How evaluation was performed

### Consider

- What are the input? Course material, ...
- What activities? Select tools, analyse case projects...
- In which order? Explore cases & tools, select tools, design FW ...
- Motivation for design choices, e.g. choice of tools, framework factors

Bonus: discuss limitations of results, validity, e.g. recommendations, framework

## Assessment (see project description for details)

G		Bonus points
Form		
Correct use of IEEE template & page limitations, Top-Down structure, good and clear language	<i>Excellent</i> top-down flow of text incl <i>Intro moves</i>	2
Work & Content		
<b>All content</b> requested in project description including <b>SPA reviews</b>	<i>Excellent</i> descriptions of <i>SPM, case projects, method incl limitations. &gt;2 scientific references</i> in Introduction.	5
<b>Evaluation framework</b> appropriately designed including choice of properties and measurements to include. Clear reporting of the <b>evaluation results</b> .		
Well-presented <b>tool recommendation</b> per project type, motivated by evaluation results.	Tool <i>recommendations are excellently motivated</i> and presented based on the evaluation results and clearly <i>connected to project characteristics</i> .	2
Oral presentation		
Clear and understandable, and within time.	<i>Excellent</i> . Use of <i>rhetorical model</i> .	1

## Student Peer Assessment



ELSEVIER

Journal of Second Language Writing 18 (2009) 30–43



To give is better than to receive: The benefits of peer review to the reviewer's own writing

Kristi Lundstrom, Wendy Baker \*

Department of Linguistics and English Language, Brigham Young University, Provo, UT, USA

### Abstract

Although peer review has been shown to be beneficial in many writing classrooms, the benefits of peer review to the reviewer, or the student giving feedback, has not been thoroughly investigated in second-language writing research. The purpose of this study is to determine which is more beneficial to improving student writing: giving or receiving peer feedback. The study was conducted at an intensive English institute with ninety-one students in nine writing classes at two proficiency levels. The “givers” reviewed anonymous papers but received no peer feedback over the course of the semester, while the “receivers” received feedback but did not review other students’ writing. An analysis in the gains in writing ability measured from writing samples collected at the beginning and end of the semester indicated that the givers, who focused solely on reviewing peers’ writing, made more significant gains in their own writing over the course of the semester than did the receivers, who focused solely on how to use peer feedback. Results also indicated that givers at the lower proficiency level made more gains than those at higher proficiency levels and that slightly more gains were observed on global than local aspects of writing.

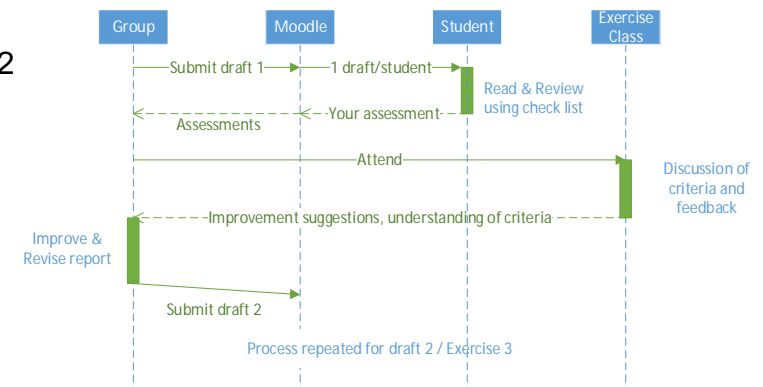
© 2008 Elsevier Inc. All rights reserved.

Keywords: Second-language writing; Peer review; Peer editing; Peer feedback

## Project Deliveries vs Exercises\*

- 50 h before Ex2
- 2 h before Ex2
- At Ex 2

- 50 h before Ex3



Check lists with criteria detailing grading criteria

\* See course schedule

# Moodle: Register for course

- Go to moodle.cs.lth.se
- Log in with stillD
- Register for ETSF01 course with group key: **ETSF01-in**, where in = Group name  
Example: **ETSF01-a1** for members of project group A1

NOW!  
27-29th March

Dashboard ► Courses ► Kurser 2016/2017

The screenshot shows the Moodle interface for the 'Kurser 2016/2017' course category. The left sidebar contains a 'NAVIGATION' menu with links to Dashboard, Site home, Site pages, My courses, and Courses. Under 'Courses', 'Kurser 2016/2017' is expanded, showing a list of courses: EDAF15, EDAF35, ETSA02, ETSF01 SPM, EDAA35, and EDA501. The main content area shows 'Course categories:' with 'Kurser 2016/2017' selected. Below this is a search bar and a list of courses: 'Algorithm implementation', 'Operating systems', 'ETSA02 Programvaruutveckling - metodik', 'ETSF01 Ingenjörprocessen för programvaruutveckling - ekonomi och kvalitet', and 'Utvärdering av programvarusystem'.

HOME PAGE MY MOODLE COURSES - MOODLE HELP -

Dashboard ► Kurser 2016/2017 ► ETSF01 SPM ► SPA I (Exercise 2 - Wed 5 April) ►

The screenshot shows the Moodle interface for the 'SPA I (Exercise 2 - Wed 5 April)' course. The left sidebar contains a 'NAVIGATION' menu with links to Dashboard, Site home, Site pages, Current course, and My courses. Under 'Current course', 'ETSF01 SPM' is expanded, showing 'Participants', 'Badges', 'General', and 'SPA I (Exercise 2 - Wed 5 April)'. Under 'SPA I (Exercise 2 - Wed 5 April)', 'SPA I: Class A (kl 10-12)' is selected, showing 'My submission', 'SPA I: Class B (kl 10-12)', 'SPA I: Class C (kl 13-15)', 'SPA I: Class D (kl 15-17)', and 'SPA II (Exercise 3)'. The main content area shows 'SPA I: Class A (kl 10-12)' and 'Submission phase'.

1 pdf file per group  
named  
<Group>\_Draft1.pdf

SPA I: Class A (kl 10-12)

Submission phase

The screenshot shows the 'Submission phase' for 'SPA I: Class A (kl 10-12)'. It features a progress bar with three phases: 'Setup phase', 'Submission phase' (highlighted in green), and 'Assessment phase'. The 'Submission phase' includes a 'Submit your work' button and a note: 'Open for submissions from Thursday, 23 March 2017, 1:25 pm (yesterday)'. Below the progress bar, there are 'Instructions for submission' and a 'Your submission' section.

Instructions for submission ▼

Submit draft 1 of your project report, named An\_Draft1.pdf where n is your group number

Deadline: 52 hours prior to exercise 2 for class A (3rd April 8.00)

Your submission ▼

You have not submitted your work yet

Start preparing your submission

COURSES - MOODLE HELP -

The screenshot shows the Moodle interface for the 'SPA I (Exercise 2 - Wed 5 April)' course. The main content area shows the 'SPA I (Exercise 2 - Wed 5 April)' title and a description: 'Peer assessment of Draft 1 of project report. Each project group (X1-X6) submits their project report, each student receives one assignment to review another group's report.' Below this are three numbered steps: 1. Each group submit draft 1 of their project report (named Xn\_Draft1.pdf, where Xn is group name, e.g. A1). Deadline: 52 hours prior to exercise class 2. 2. Teacher assigns a report to each student for review 3. Read the assigned report and provide feedback in Moodle according to the criteria. Deadline: 2 h prior to exercise class 2. Below the steps are two notes: 'NOTE: We recommend using the SPA I review list (see course web page) while reviewing to avoid Moodle timing out, then enter your feedback into Moodle.' and 'NOTE: Active reviewing is required of each student as part of the project assignment.' Below the notes are four links: 'SPA I: Class A (kl 10-12)', 'SPA I: Class B (kl 10-12)', 'SPA I: Class C (kl 13-15)', and 'SPA I: Class D (kl 15-17)'. The right sidebar contains a 'SEARCH FOR' section, a 'LATEST ANN' section with a link to 'Add a new topic', an 'UPCOMING' section with a link to 'Go to calendar...', and a 'RECENT ACT' section with a link to 'Activity since F'.

ssen: Ekonomi & Kvalitet

se 2 ► Student Peer Assessment (SPA) I ► Redigerar formulär för bedömning/värdering/betygssättning ► Förhandsgranska

Student Peer Assessment (SPA) I

Formulär för bedömningar/värderingar/betygssättningar

Aspect 1

IEEE template incl formatting of pages, title, names, headings, paragraphs, table and figure captions, references, etc.

How is the template used? Correctly including references in text and in reference list?

Comment\*

Aspect 2

Report structure

Does the report contain (at least a draft of)

- Title and authors including group name
- Abstract
- Introduction
- Method
- Case projects
- Evaluation framework
- Tool evaluation and improvements
- Tool recommendations per project type
- Conclusions
- References

Comment\*

1. Read the report first
2. Consider the criteria / aspects
3. Re-read relevant parts of the report
4. Provide constructive feedback

Criteria also found on course page – Ex2

## Goal-Question-Metric (GQM)

Method for designing SW metrics to assess goal fulfillment

1. Define what the **goals** are, e.g. for tool support of planning
2. Define **questions** that determine if goal is met
  - Refine goals
  - Learn about progress towards goals
3. Define **metrics** (**== factors in your evaluation FW**) that
  - Answer / measure each question
  - Determine if goal is achieved

P1: V.R. Basili, Lindvall, Regardie, Seaman, Heidrich, Münch, Rombach, Trendowicz, "Linking Software Development and Business Strategy through Measurement", IEEE Software, April 2010, pp. 57-65

## Evaluation Factors for Effort Estimation

- Goal: When is Effort Est successfully supported by a tool?
  - Accurate estimates for android changes [Porting]
  - Estimates based on cost for previous similar projects [both]
  - Compile detailed estimates from dev teams [Porting]
  - Agreed estimates at sprint-planning [app]
  - In combination with duration and optimal resource allocation
- Question that determine if that goal is achieved?  
Measurement?
  - Can relevant factors be used for analogy-based estimation?  
Type of support: none, pre-defined factors, customizable factors
  - Is poker planning supported?  
To what degree: no, via app,
  - Can estimates be requested to be detailed by other users?

## Evaluation Factors for Effort Estimation

When is Effort Estimation successfully supported by a tool? -> **Goal**

What **question** help determine if that goal is achieved?

How assess/answer Q? -> measurement = factor + scale

## Evaluation Framework

Identify suitable factors to evaluate for 5 SPM areas

- Activity planning
- Effort estimation
- Risk management
- Resource allocation
- Monitor & control execution

+ Quality aspects, e.g. usability (changes), performance, capacity

Define measurements with scales for each factor

ENSURE a good mix!

- & scales**
- ENSURE a good mix!

- Support for estimation techn: {analogy, COCOCO, .}
- Degree of analogy: {None, Simple, Advanced}



Lund University / Faculty of Engineering/ Department of Computer Science / Software Engineering Research Group

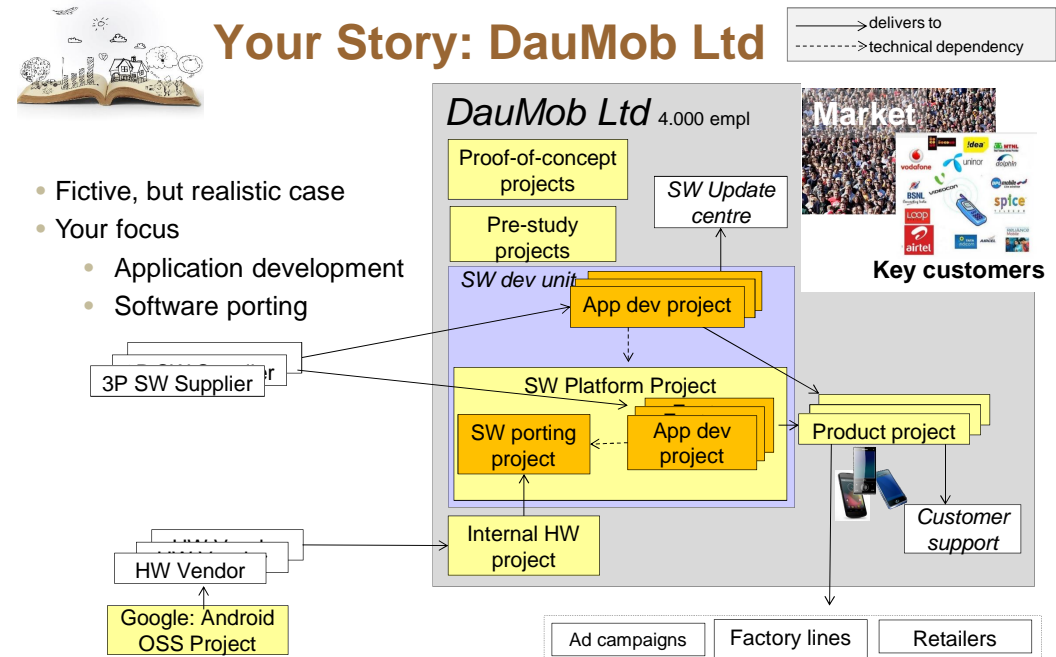
## A hand-drawn illustration of a landscape scene, including a sun, clouds, a house, trees, and figures, appearing to be drawn on a piece of paper or a book cover.

- Large company producing consumer devices
- Many different project types
- Your focus: projects for
  - Ø software porting
  - Ø application development

**Case-based teaching**

- active learning
- Read/discuss

- active learning strategy
- Read/discuss/analyse complex, real-life scenarios



## Case: SW Porting Project



- § Ports app level SW to new HW platform incl company patches to Android OSS
- § **Lead time** is important: 4-7 months until main release
- § Dependencies
  - Delivers to SW Platform & Product projects: plans and HW platform releases
  - Uses external HW components
- § Process
  - § **Phase-based with 3 increments:** 2 pre-releases and 1 main release
  - § **Phases:** design & planning, implementation, system testing incl certification, maintenance
  - § **Coordinating roles:** project managers, requirements coordinator, senior architect, integration & configuration manager (CM), system test lead, quality coordinator
  - § **Software area team (20-25 teams):** 1 team leader, 1 reqts/product owner, 1 area architect, 1-3 developers (code + functional testing)

## Case: Application Development

- § **New or updated application**, e.g. TV integration feature, health app, social game
- § Requirements: customer-specific apps or driven by market needs
  - high-level reqs, details are usually left to developers
- **Lead time can be critical:** 9w – 2 years
- § **Dependencies**
  - Deliver to SW Platform &/ Product prjcts **200-400 app projects / SW Platform**
  - May use 3-party software & have dependencies to HW and/or Android OSS
- § Process
  - **Scrum w initial pre-study period** then **iterations (sprints)** w coding & testing
  - Product owner (customer repr): approves scope incl reqts changes
  - Sponsor (line manager): ensure sufficient resources
  - Scrum master (PM): support team w planning incl risks, monitor & report status to SW Platform project
  - **Team of 1-20 devs:** detail HL reqts w product owner, code and test agreed reqts
  - **Dedicated tester:** plan larger test effort & coordinate w SW platform system test lead



## Activity Planning: Software Porting



Scope: Enable SW platform for Lolipop release

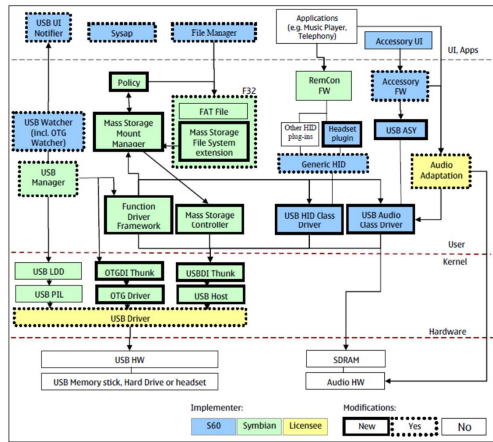
- **Identify impact of new release on existing SW modules**  
E.g. Security, Audio, Video, Touch, Sensor, Local Connectivity, Device Drivers, Location Services.  
Who: SW Architect and Requirement Engineer  
Output: New Lolipop features and impact modules
- **Identify activities per feature and SW module**  
Who: Project manager & Tech area SW Arch  
Output: An activity plan per SW module and per related new feature. MS Project task list is an example on a used activity tool.

## ACTIVITY PLANNING



# Software Porting: Activity Planning

## Example: USB, including OTG



Key use cases are external media (e.g. HDD), Keyboards (HID), and Audio over USB. Most of the components are supplied. The complex component is the audio adaptation.

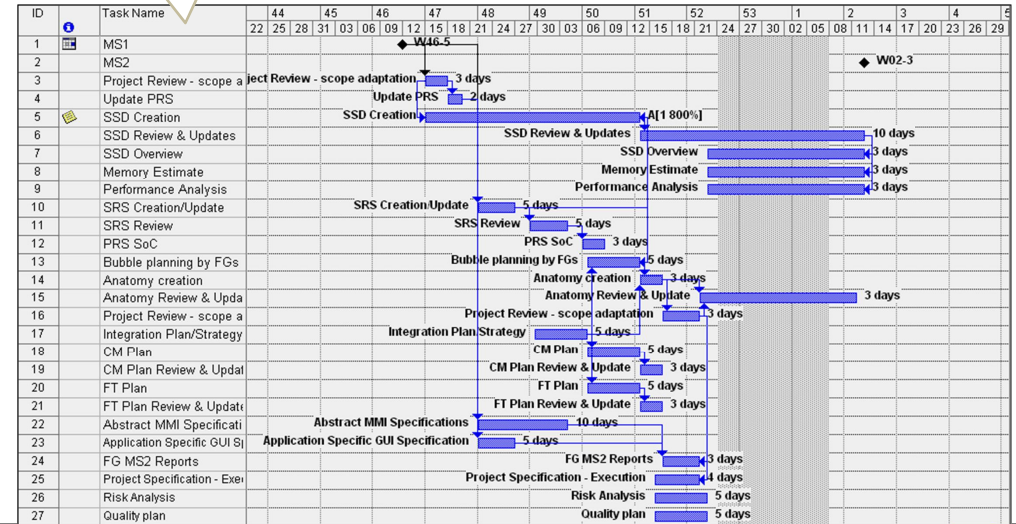
- Activities identified based on **impact** on software architecture components
- Based on documentation of **changes** in new Android cookie
- First by SW architects (Top) then refined by Software engineers per technical area (Down)

Tasks
USB adaptation
Verification
OTG (OnTheGo)
USB Audio (optional)

# Software Porting: Activity Planning

Activities and Work products

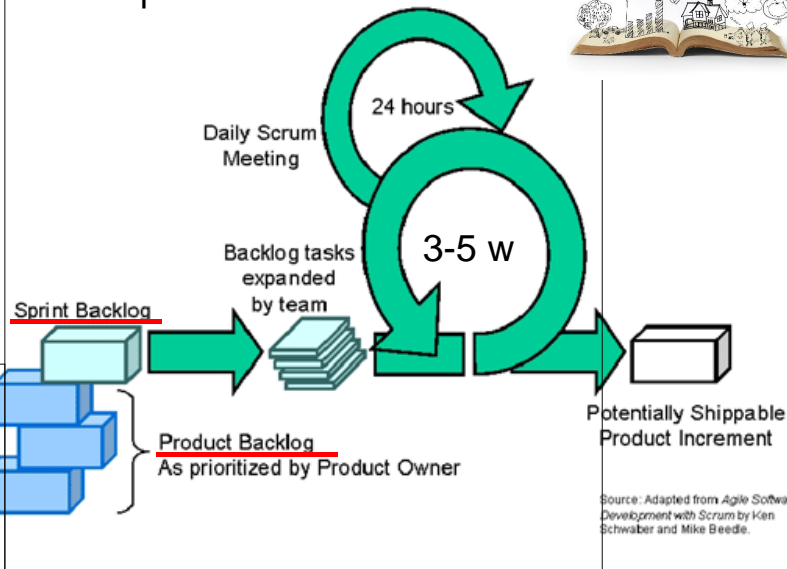
Pre-study Status, Feasibility planning



## Activity Planning: App Projects

Develop

- Prepare
- New functionality
  - Tech dependencies
  - Other tasks



## Activity Planning: Application Project

Scope: A new version of a health app is to be developed

### - Before implementation begins: Prepare backlog

- Product Owner (acting as the customer) defines the **new functionality** as user stories and prioritizes these in the product backlog.
- SW architect identifies **technical dependencies** between the user stories and updates the backlog order.
- project manager adds additional **high-level activities required by the process**, e.g. sketch UI interaction flow

- **For each sprint:** SW architect and software developers identify tasks (activities) for the most prioritized user stories. These are ordered and added to the sprint backlog.



## EFFORT ESTIMATION

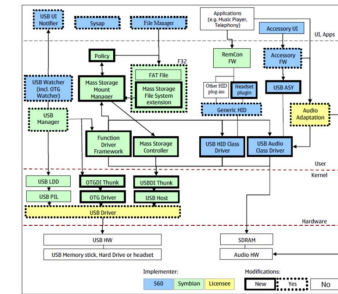
## Application Dev Project: Effort estimate

§ Pre-study phase: **Expert judgement using analogy** by SW architect

- Considers overall impact of new requirements
- Considers experience of team: individually & together
- Dev phase. Per sprint & User story: **Planning poker** by development team members



## SW Porting: Effort estimation



### Rough estimate

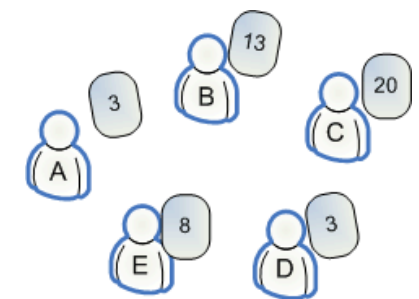
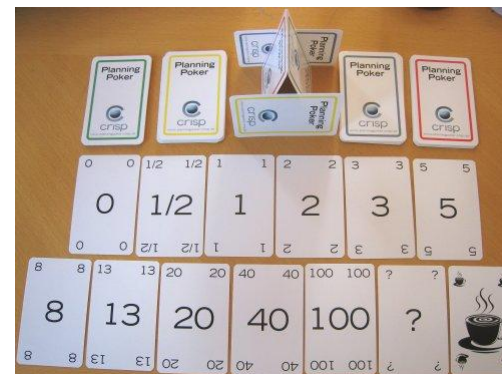
- by Senior architect
- Analogy – cmp previous
- Expert judgement, main impact

### Detailed estimate

- by SW area teams
- Expert judgement of tasks

Task	Effort (mw)
USB adaptation	3
Verification	3
OTG	4
USB Audio (optional)	5
Total	15

## Planning Poker (Story points)





## RISK MANAGEMENT

### Risk identification & analysis

Brainstorming with core PM team

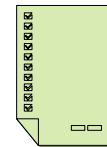
- Software area teams contacted, if needed



### Risk list



### Risk monitoring



Other purposes

- Bring project team together
- Establish "us"
- Discover product needs

Select 5-10 top risks

Monitor, e.g.

- used as agenda at project meetings
- identify actions to mitigate these risks

## Case Example: Estimating Risk Exposure



Risk exposure = Severity \* Probability = Risk priority

Risk	S	P	R	Action
External deliveries No 10 may be late and of bad quality.	5	5	25	1) Plan a focus meeting and review the work break down and update the resource estimates. 2) Check if we can include penalties in contract
Lack of resources in area No 2	4	5	20	1) Make an analysis with Current, Minimum and recommended resources within the area.  2) Ensure that project needs are taken into account in line planning (through steering group)
Graphics performance too poor	5	3	15	1) Request to configure without Graphics accelerator. 2) Perform performance test and increase resource allocation.

## Case Example: Risk prioritization/exposure



Severity	Schedule Delay on Launch (time)	Functionality/ Performance (scope)	Perceived quality (scope)
1	1-2 w	Reduced performance on a non key functionality	Customer notices reduced performance on a non key functionality
2	3-4 w	Drop of a non key functionality	Customer annoyed on quality of non-key parameter
3	1-2 m	Reduced performance on a key functionality	Customer annoyed on quality of key parameter
4	2-3 m	Drop of a key functionality	Customer complaint
5	>3 m	Drop of several key functions	Product return, non-recommendation

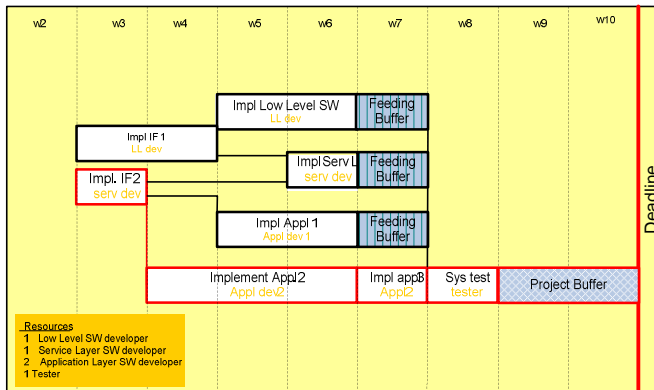
Probability	
1	<20 % probability that risk will occur
2	20-40 % probability that risk will occur
3	40-50 % probability that risk will occur
4	50-60% probability that risk will occur
5	>60 % probability that risk will occur

General prio per project type

- SW Porting: time
- App Dev: scope -funct

# SW Porting: Risk management

## Securing the critical chain\* with buffers



\*longest chain of activities consider task & resource dependencies

= critical path + resource limitations

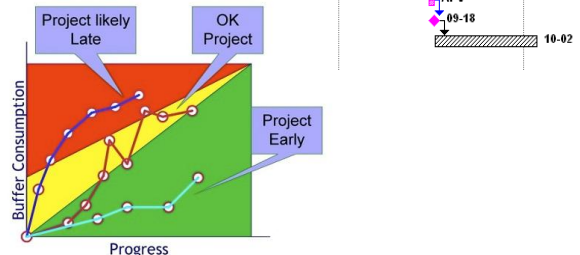
50% / 90% estimates of each task.

Duration = 50% estimates, The rest (51-90%) in buffers

- Project buffer =  $\text{Sum}(t_{90}-t_{50}) / 2$  for the **tasks in the critical chain**
- Feeding buffer =  $\text{Sum}(t_{90}-t_{50})/2$  for **chain connecting** in to the critical chain

# SW Porting: Executing a Critical Chain Plan

Task Name	Focus Dur	Low-Risk Dur	Start	Finish	Pro
1 Conversations	33 days	0 days	Fri 07-08-17	Tue 07-10-02	###
2 Dev Presentation	10 days	20 days	Fri 07-08-17	Thu 07-08-30	###
3 Dev Framework	5 days	10 days	Tue 07-08-21	Mon 07-08-27	###
4 FBDev Framework-3	3 days	0 days	Tue 07-08-28	Thu 07-08-30	###
5 Dev apv	5 days	10 days	Tue 07-08-21	Mon 07-08-27	###
6 FBDev apv-4(APV & I	3 days	0 days	Tue 07-08-28	Thu 07-08-30	###
7 APV & BF	10 days	20 days	Fri 07-08-31	Thu 07-09-13	###
8 Integration	1 day	2 days	Fri 07-09-14	Fri 07-09-14	###
9 APV sanity	1 day	2 days	###	###	###
10 Delivery	0 days	0 days	Tu		
11 PBDelivery-8	11 days	0 days	Tu		



## Critical chain approach (cont.)

- “Critical chain” also considers resources
- Put a project buffer at the end of the critical chain with duration 50% of sum of comfort zones of the activities on the critical chain
- During project execution monitor how much of the buffer that has been used
- Supported in tools, e.g. through add-on to MS Project

## Executing & Monitoring CC plans

- Principle: focus your efforts - “multitasking i evil”
  - No **chain** of tasks is started earlier than scheduled, but once it has started is finished as soon as possible
  - This means the activity following the current one starts as soon as the current one is completed, even if this is early – the relay race principle
- Fever charts are used to monitor progress and catch tasks at risk

## Application Dev Project: Risk Management



- Informal and integrated in Scrum process
- Depends on individuals

### Pre-study:

- Product owner performs risk identification & prio
- Affects backlog prio & communication with team

## Application Dev Project: Risk Management

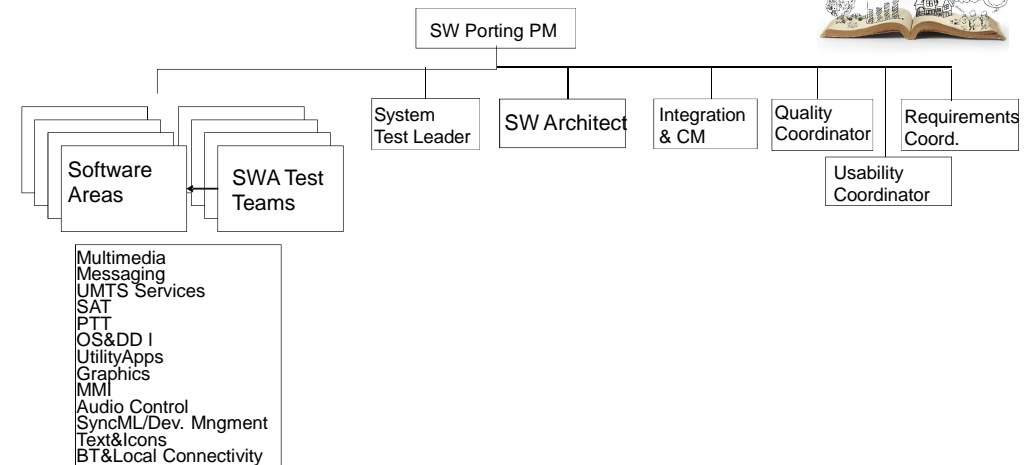


- Risks managed continuously, during development
- **Transparency** incl continuous dialog with customer
  - Risks **discussed in planning poker** and included in estimates
  - If too much unknown, a **"spike"** can be performed
  - Hindrances mentioned at daily stand-up **meetings**



## ORGANISATION AND RESOURCE ALLOCATION

## SW Porting Project Organisation



## SW Porting Project: Resource allocation

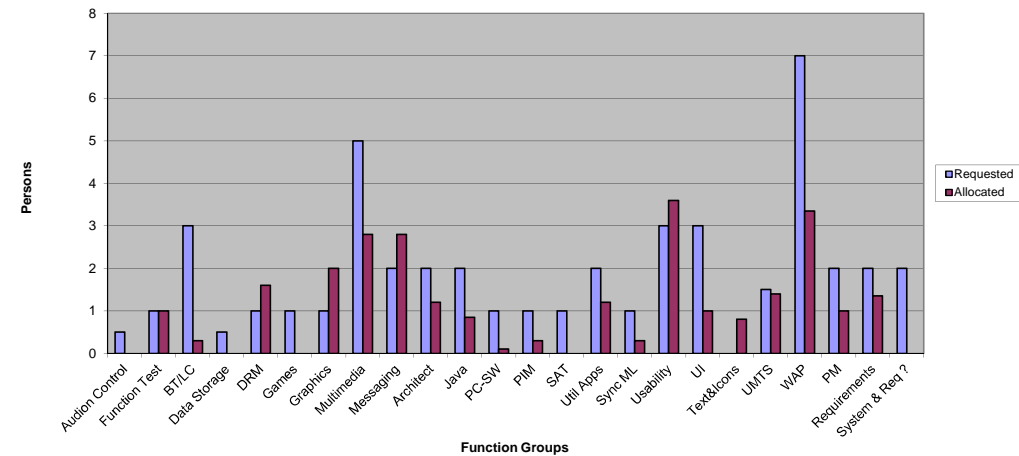


Resource Request per	200602	200602	200602	200603		
Function Group	200602_REQ	200602_ALL	Diff	200602_REQ	200602_ALL	Diff
L_PG_A-DRM L_PG_A-DRM	1,7	1,9	-0,2	1,7	1,6	0,1
L_PG_BTLC L_PG_BTLC	2,65	2,4	0,25	2,1	1,85	0,25
L_PG_CORE L_PG_CORE	3	2,7	0,3	2	2,3	-0,3
L_PG_GAMES L_PG_GAMES	3,75	3	0,75	2,75	2,75	0
L_PG_GFX L_PG_GFX	3	3	0	3	3	0
L_PG_IMM L_PG_IMM	6,6	6,6	0	4,4	4,4	0
L_PG_MESSA L_PG_MESSA	6	7,75	-1,75	5	4,5	0,5
L_PG_OAFL L_PG_OAFL	4,5	5	-0,5	3,5	3,75	-0,25
L_PG_PCSW L_PG_PCSW	1	1	0	1	1	0
L_PG_PIM L_PG_PIM	1,75	1,75	0	1,75	1,75	0
L_PG_PM-SWL L_PG_PM-SWL	4	3	1	4	3	1
L_PG_SAT L_PG_SAT	1	1	0	1	1	0
L_PG_SPEC L_PG_SPEC	3,2	3,2	0	3,5	2,9	0,6
L_PG_SVER1 L_PG_SVER1	8,5	7,85	0,65	8,2	8,05	0,15
L_PG_SVER2 L_PG_SVER2	12,55	13,05	-0,5	12,9	12,6	0,3
L_PG_SWARCL L_PG_SWARC	0,2	0,2	0	0,2	0,2	0
L_PG_SWPRO L_PG_SWPRO	1	1	0	1	1	0
L_PG_SYDEV L_PG_SYDEV	4	3,1	0,9	4	1,8	2,2
L_PG_UIAPP L_PG_UIAPP	3,5	3,25	0,25	3,5	1,25	2,25
L_PG_UIDES L_PG_UIDES	0,8	1,05	-0,25	0,8	1	-0,2
L_PG_UIGUI L_PG_UIGUI	4	3,41	0,59	4	2,91	1,09
L_PG_UISPC L_PG_UISPC	2	1,4	0,6	2	1,4	0,6
L_PG_UITXT L_PG_UITXT	5	3,7	1,3	5	3,7	1,3
L_PG_UMTS L_PG_UMTS	4,5	2,75	1,75	4,25	1,75	2,5
L_PG_VERCOL L_PG_VERCO	2	2	0	2	2	0
L_PG_WAP L_PG_WAP	2,75	5,2	-2,45	2,75	4,5	-1,75
SUM	91,25	87,36	3,89	84,6	73,36	11,24

## SW Porting: Resource allocation is continuous



November - Requested versus Allocated

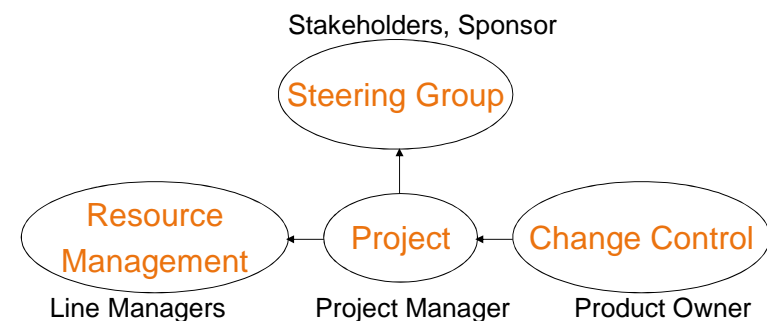


## SW Porting: Resource Allocation



- PM requests resources based on estimated activity plan
- Line managers allocate resources to different projects
- PM considers diff
  - Was the request right?
  - If overallocated, talk to managers. Do they info project is missing?
  - If underallocated, do consequence analysis and consider alternative plans &/ arguments for more resources. Escalate to steering committee.

## Project Steering: Software Porting Project



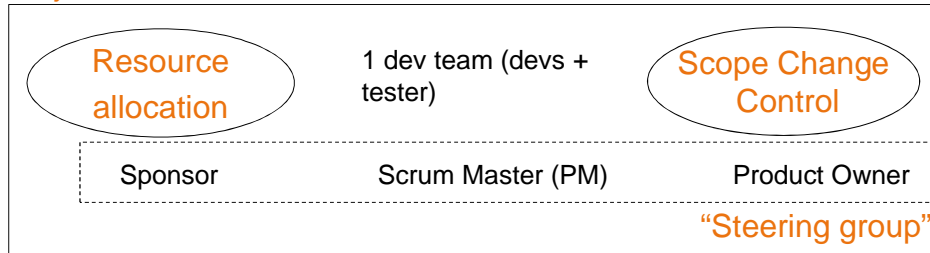
# Project Steering: Application Dev Project



Initial scope &  
resources

then primarily self-  
governance

Project



# Application Dev Project: Resource Allocation



High-level allocation of team members, then **self-governing teams**.

- Overall effort estimate from pre-study used to request & allocate team
- During iterations/sprints tasks are "pulled" by team members according to prio order. **No PM allocation of tasks.**
- For each sprint planning, the team capacity is calculated based on team members availability & previous team velocity
- Problems, e.g. with rate of progress, discussed within the "steering group", i.e. Product owner, Scrum Master and Sponsor

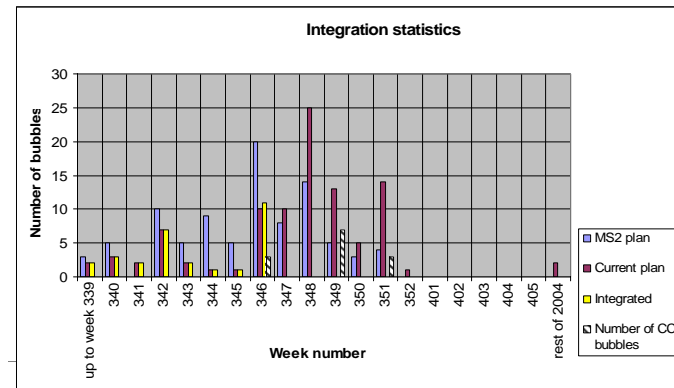
# SW Porting Project: Monitor and control



Weekly status report collected by PM fr teams & tracking systems

- Progress relative delivery scope & timeline
- Software quality status (performance)
- Risks and Actions

Presented at project meeting &  
to steering group & sponsor



## Summary

The project is in the execution phase and the project includes 97 SW deliveries (bubbles) in the Anatomy. 28 SW deliveries (bubbles) are now delivered. A checkpoint is scheduled in week 48.5 in order to summarize the status and to decide how to move forward

## Critical areas are:

1. Deliveries are pushed forward, see below integration statistics.
2. Graphical performance. The graphical performance is only 25% of the expected performance. Root cause analysis is ongoing
3. Quality problems with the FM-Radio RDS chip. Re-planning is ongoing.

## MONITOR & CONTROL

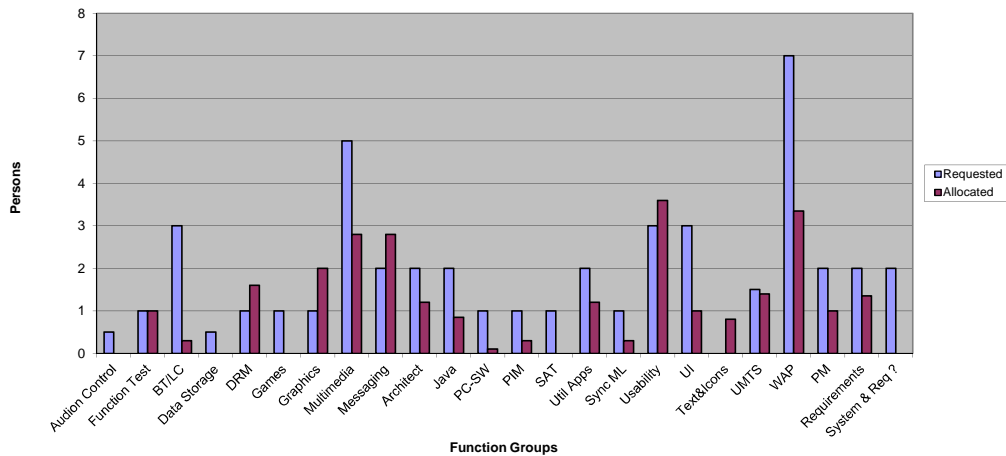


## SW Porting Project: Monitor & Control

Resource allocation monitored on a monthly basis



November - Requested versus Allocated



Lund University / Faculty of Engineering/ Department of Computer Science / Software Engineering Research Group

## SW Porting Project: Monitor & Control of **COST**



- Reported once a month & at checkpoints to steering group
- Extracted from internal systems
- View progress, not just expenditure

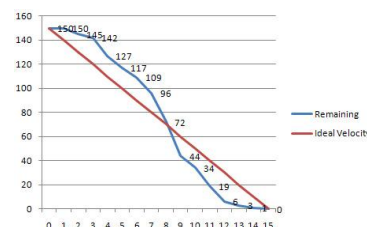
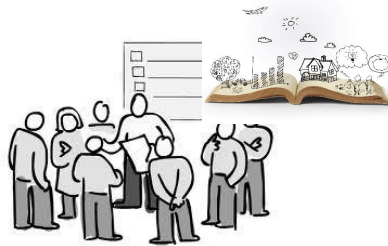
### Cost monitoring

SYSTEM Project Order & Cost (KEUR)	Total Actual	Total Forecast	October Actual	October Forecast
Man Months	92	130	19	27
Labour hours	12 989	18 302	2 685	3 779
Labour costs	1 348	1 830	290	378
Material/Consumables	100	60	10	20
Travel & Living	11	39	3	5
Consultants	10	20	5	7
Misc	2	5	1	2

Lund University / Faculty of Engineering/ Department of Computer Science / Software Engineering Research Group

## Application Dev Project: Monitor & (Team) Control

- Regular feedback & knowledge share
  - Daily stand-up meetings
  - Sprint demos & planning, sprint retrospectives (SPI)
- Burn-down charts used to monitor progress & "remaining work"
- Dependant projects
  - Status reporting delivered to SW Platform & Product projects
  - Status reports received from, e.g. SW porting project. Info on dependent functionality & deliveries, considered in sprint planning as part of backlog prioritization.



Lund University / Faculty of Engineering/ Department of Computer Science / Software Engineering Research Group



## SW PROCESS IMPROVEMENT

Lund University / Faculty of Engineering/ Department of Computer Science / Software Engineering Research Group

## SW Porting (Trad)

- Post-project meeting: lessons learnt, postmortem
- Lean Six Sigma improvement projects
- Driven by line mngement

## App Dev (Agile)

- Sprint retrospectives

- Driven by team

