Programvaruutveckling för stora system

Martin Höst
course leader, lectures, reviewer, exercises

Rasmus Ros
labs, lectures, design expert

Johan Linåker
reviewer, requirements expert, exercises

Sergio Rico
labs

http://cs.lth.se/ETSN05

Welcome!

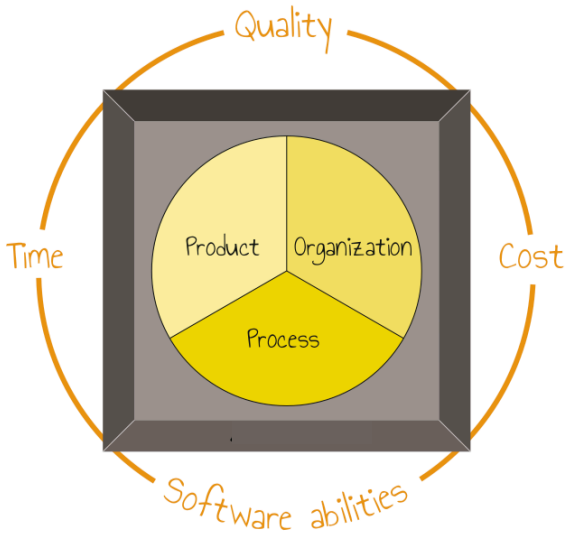
Part one:

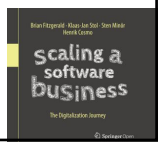
- introduction
- course/project objectives
- course programme
- course/project overview
- project roles
- assessment

Part two:

- the product
- the task
- development model

Effects of scaling





Big projects

- Requires:
  - "Software Engineering" – **not** "Software Handicrafting"
- Implies:
  - Collaboration in projects – roles and responsibilities
  - Defined way of working – processes
  - Documented work – specifications, plans, design documents

Course/project objectives

Prepare you for large-scale, industrial software development in teams.

Knowledge of:

- development processes
- project management
- central concepts from large-scale development

Experience with:

- problems in large projects
- possible solutions to problems
- one particular project role (hands-on)
- other project roles (collaboration, observation, reflection)

Software Engineering is Teamwork!

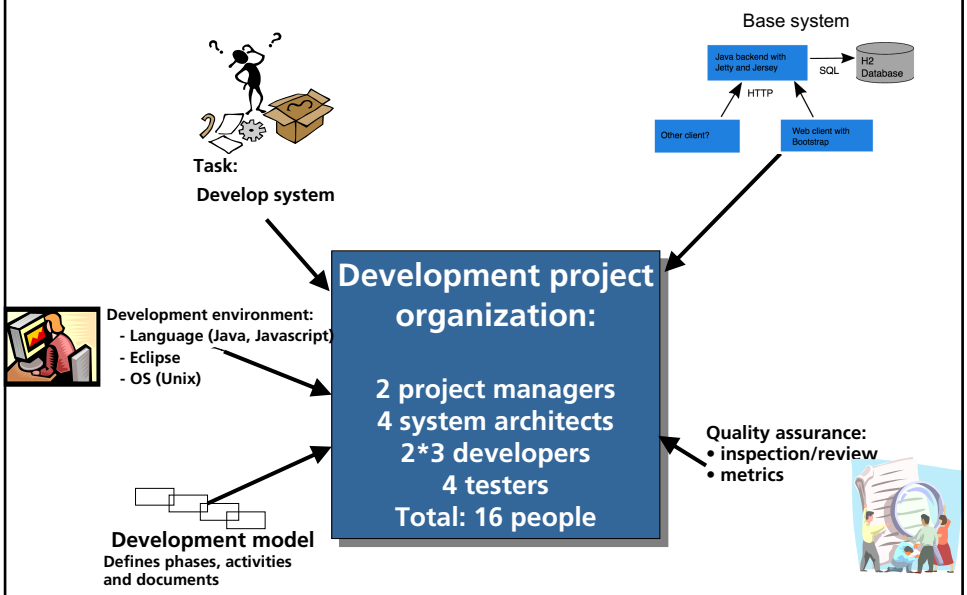
Leadership
- **Goals**
- **Plans**
- **Commitment**
- **Assessment**

Ambitions
Norms
Tolerance
Professionalism
Communication
Reflection
…

Pedagogical Approach

- Simulate a real project in a typical large software organization (**organzation**)
- Apply approaches used in current software engineering practice, such as project management, process model, milestones, reviews, etc. (**process**)
- Focus on organizational challenges while reducing technical risks, although realistic product (**product**)
- Learning by doing: Case-based learning

Development project organization



Support

Compendia:
- project guide (PH)

Lectures (4+1)

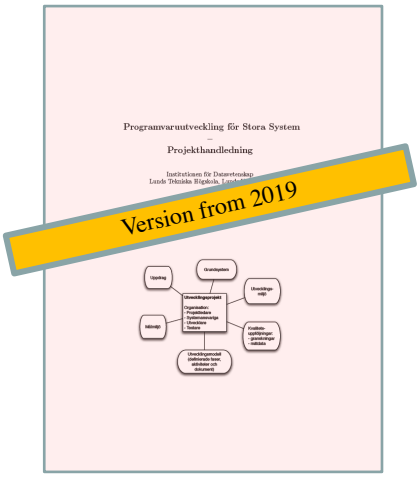
Exercises (2)

Labs (2)

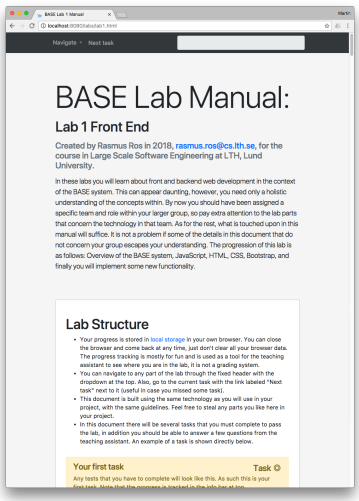
Experts (3+1):
- requirements
- design
- test
- Section manager

Reviews (2+1)

Course material



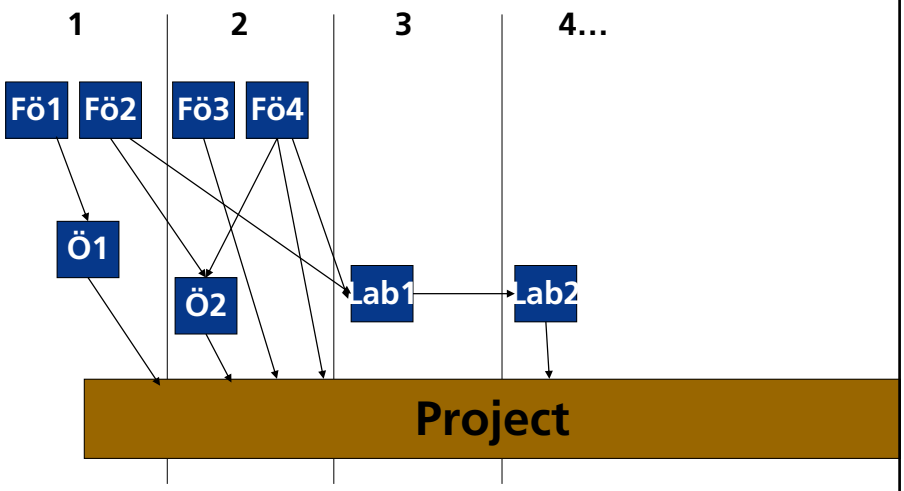
"PH"



Time plan

Akti-vitet	v1	v2	v3	v4	v5	v6	v7	v8
Fö	F1 F2	F3 F4					Fö5	
Övn	Ö1	Ö2						
Lab			Lab1	Lab2				
Gr			Gr1		Gr2			Acc-m

Overview of support sessions



Lecture 5 in week 7

- Seminar on problems in large projects
- Discussions based on your input
- You can *submit* your *discussions-points* (will be kept anonymous if preferred) as input to l ecture 5 to martin.host@cs.lth.se

The project group

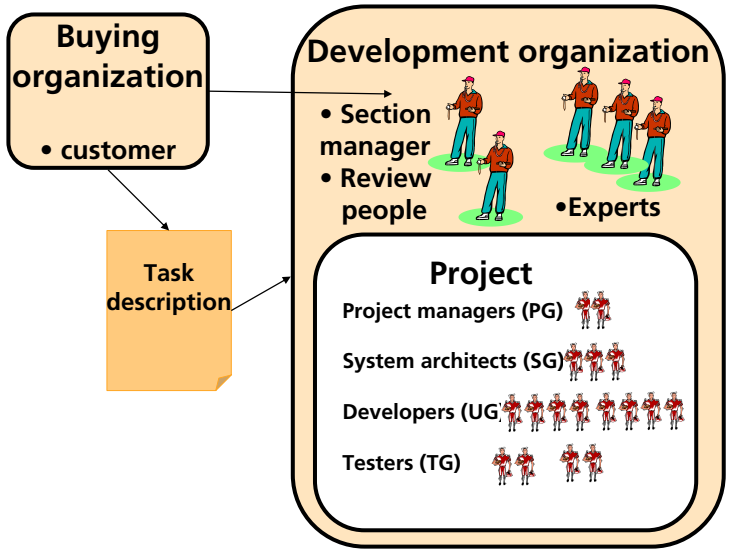
Project managers (PG)

System architects (SG)

Developers (UG)

Testers (TG)

”External” roles



Project management group (2 persons)

- responsible for the group producing a result
- plan, put together group, assign (and re-assign) tasks
- identify configuration items
- collect and report metrics data
- report to the ”Section manager”
- put together the Software Development Plan (SDP), the System Specification Document (SSD) and the Project Final Report (PFR)

Kick-off project managers:
Thursday 12:15

System group (~ 3+1 persons)

• ”architecture group” - responsible for co-ordinating the technical work
• handle the base system and interfaces
• responsible for program library and error- and change reports – competence on version control, configuration management
• communicating with developers and taking part in development
•put together the
  Software Requirements Specification (SRS),
  Software Top Level Design Document (STLDD)
  Software Detailed Design Document (SDDD)
• time reporting

Developer group (~3*2 persons)

• develop the requested new functionality
• incl. unit test
• communicating with system group and testing group
• produce parts of the
  Software Requirements Specification (SRS),
  Software Top Level Design Document (STLDD)
  Software Detailed Design Document (SDDD)
• have an own competence area (back-end, front-end, algorithm)
• time reporting

Test group (~3+1 persons)

- responsible for testing
- create test descriptions / automation and carry out testing
- communicating with development group (and supporting), participate in unit testing (experts on automated testing)
- produce the
  - Software Verification and Validation Specification (SVVS),
  - Software Verification and Validation Instruction (SVVI)
  - Software Verification and Validation Report (SVVR)
- responsible for co-ordinating appendices
- time reporting and error reporting
- a *test manager* must be appointed

Experts and Section manager

- Section manager -- SDP and PFR
- Requirements expert -- SRS
- Design expert -- STLDD and SDDD
- Test expert -- SVVS, SVVI and SVVR
- ”ePUSS-guru” -- web system for reporting

People from Computer Science are experts and Section manager, customer (via the Section manager) and quality evaluators (through reviews)

Open positions as project managers

We are looking for project managers!
The job requires:
- persistance, will to carry through, discipline

The jobs gives:
- experience, overview, satisfaction

Previous experience not required, but merits are appreciated.

Email application to: martin.host@cs.lth.se

No later than tomorrow at 13:15

Signing up for the project

If you have doubts about taking the course - make up your mind NOW!
- participation requires that you sign up for the project (that's how I know that you will follow the course)

Choose project group via the homepage today no later than 19:00.

Groups will be formed as soon as possible - and lists be put up on the web.
…Maybe I will have to re-arrange groups.

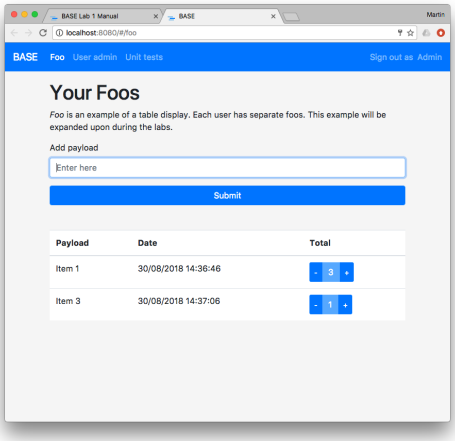
Check your email often Wednesday/Thursday, as your project managers will send information and call for the first group meeting

Sign up for exercises and labs

- Exercises: no later than tomorrow Wednesday at 18:00
- Labs: no later than next week

The product

• Base system with login and possibility to add ”Foos”



Your task

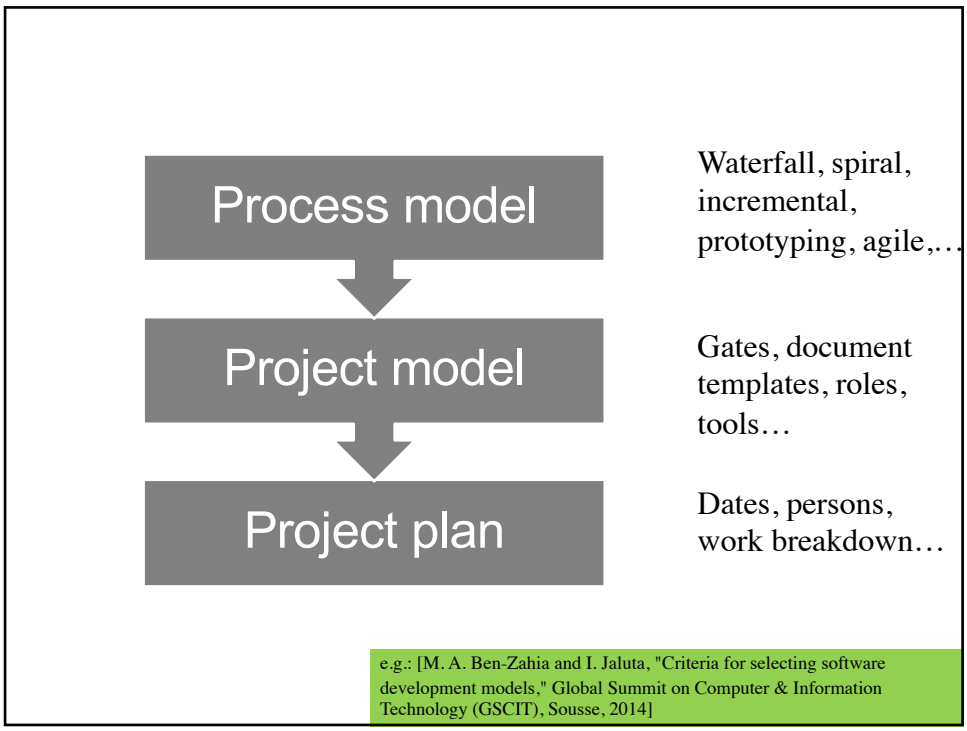
- Develop a system that help commuters find cars and passengers

Some software engineering concepts

- Process, development model
- Configuration management, change management
- Requirements engineering
- Specification, analysis, design, implementation
- Review/inspection, validation, verification and test
- Product documentation
- Project management, time planning, costs, resources
- Quality assurance
- Metrics (Metrics planning, -collection and -analysis)
- Characteristics, e.g. reliability
- Support system, CASE-tool

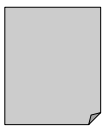
Available if you search in "LUBSearch", see lub.lu.se

e.g.: [G. O'Regan, Concise Guide to Software Engineering: From Fundamentals to Application Methods, Springer, 2017]

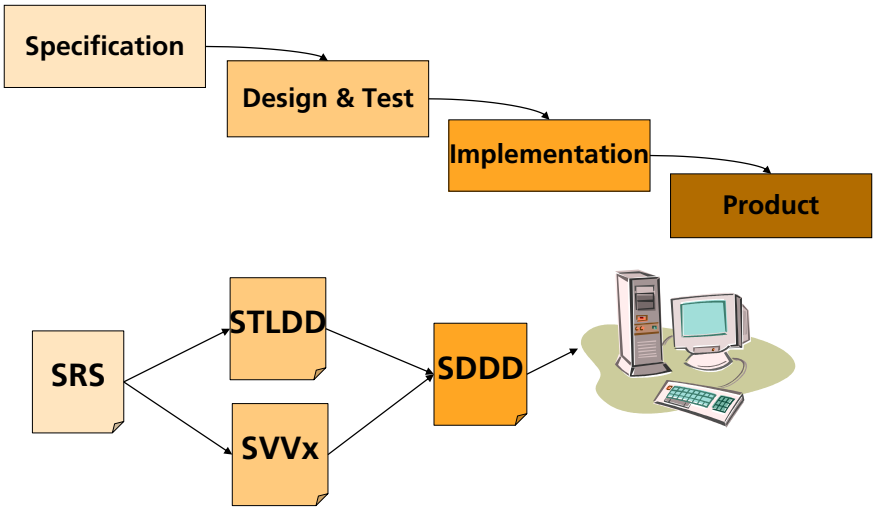


Concepts in the development model

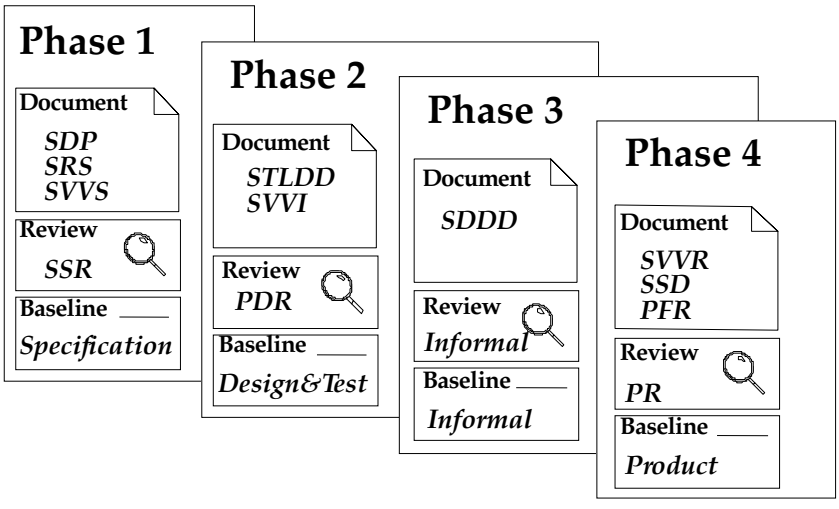
- Document
- Review/inspection
- Baseline

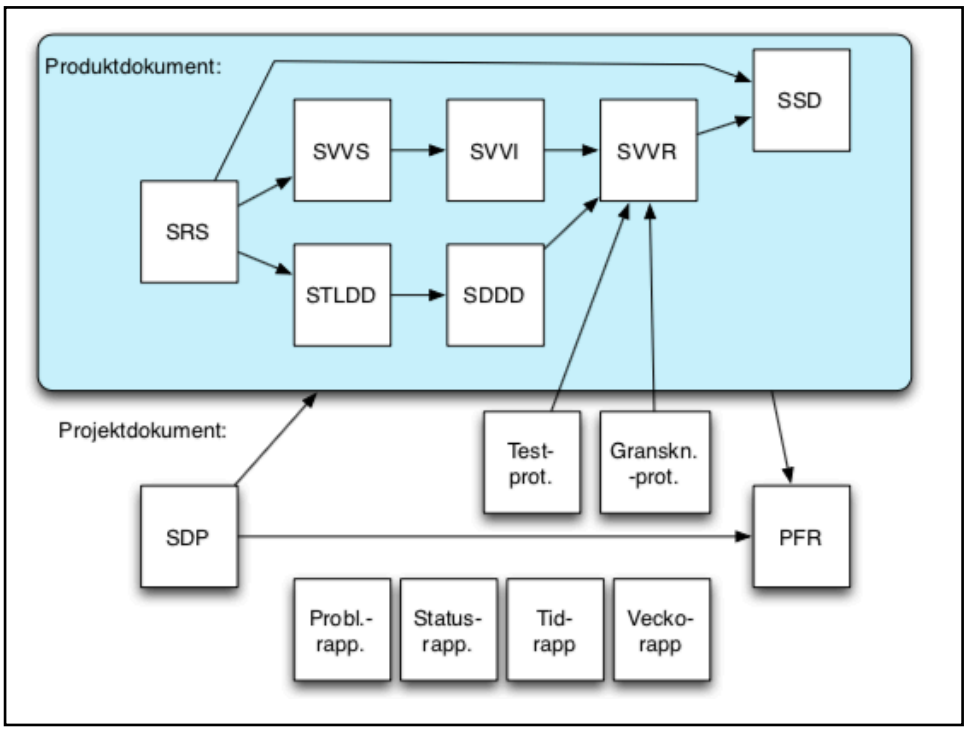


From requirements to functionality



Development model





Review/inspection

A meeting where you in a structured way check that documents are correct and of a high quality. All errors are recorded.

Two types in this project:

- ”informal”: the project’s internal quality control
- ”formal”: auditors/reviewers/inspectors/ quality assurers + the whole project group

Configuration Management (CM)

CM is a controlled way to manage development and change to systems during the whole life-cycle

Why do we need CM?

- to enable parallel work
- to know at any time what parts depend on other parts and which is the valid version
- to have a strategy for how to ”integrate” changes and corrections

Configuration unit

- A unit that can be handled as independently as possible.
- Should be identified, have a version number, and a change history

Baseline

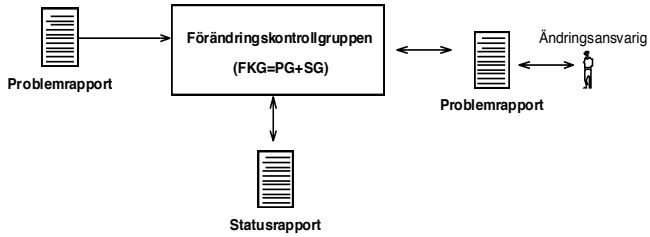
After a review and bug correction, you create a baseline whose constituent documents are ”frozen”. Subsequent changes are controlled.

Change management

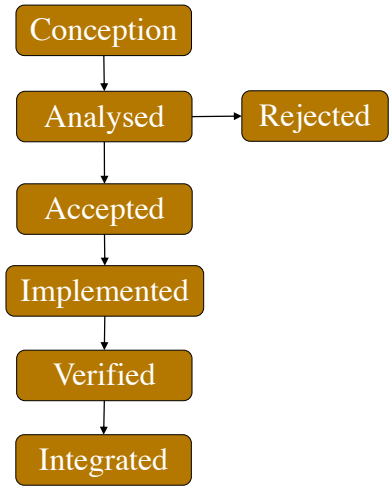
- Changes in baselines will be necessary
- All configuration units have a status and a list of changes that are being made
- The project must decide in more detail how change management should be done in the project
  - Formal procedures necessary after baseline

Change Control Board (CCB)

- make decisions about changes
- the status report provides the history for each CI
- the problem report is like a baton that is passed between the different parties involved in carrying out a change and where all decisions and actions are documented



The life-cycle of a change



Change management support documents

- Problem report – consists of problem description, assessments of problems, proposed solutions, status of changes etc.
- Status report – consists of status of all ongoing and conducted changes.
- May be done through “ePUSS” or any other formalized way

Metrics

- A basis for understanding and learning
- For every item in the SDP plan the PFR should report the actual outcome, e.g.
  - Effort in every phase
  - Effort for every document
  - Effort per activity in different phases
  - Effort per group (e.g. UGx, TG, etc)
  - …

On metrics programmes on organizational level: T. Kilpi, Implementing a software metrics program at Nokia, IEEE Software, Volume: 18 Issue: 6, 2001.

Nummer	Aktivitet	Specificering
11	SDP	Arbete med dokument med bilagor.
12	SRS	
13	SVVS	
14	STLDD	
15	SVVI	
16	SDDD	
17	SVVR	
18	SSD	
19	Slutrapport	
21	Funktionstest	Arbete med testning
22	Systemtest	
23	Regressionstest	
30	Möte	Gruppmöte, expertmöte, etc
41	Föreläsning	Inlärning, "kurstid"
42	Övning	
43	Terminalövning	
44	Självstudier	
100	Övrigt	

Aktiviteter

	Nr	Feltyp	Eventuell beskrivning
Generella	11	Stavfel, syntaxfel	
	12	Referensfel	Felaktig/saknad referens inom eller mellan dokument
	13	Standard	Följer ej standard
	14	Relevans	Innehåll ej relevant
	15	Redundans	
	16	Avsaknad	Dokument ej fullständigt
	17	Inkonsistens	Motsägelser
	18	Begriplighet	Formulering/jod onödigt svår att förstå
	19	Entydighet	Formulering mångtydig
SRS	21	Inkorrekt krav	
	22	Spårbarhetsproblem	
	25	Redundant krav	
	26	Saknat krav	
	27	Inkonsistent krav	
	28	Organisationsproblem	
	29	Entydighet	
	31	Verifieringsproblem	
SVVS,	41	Inkorrekt test	
SVVI &	42	Spårbarhetsproblem	
SVVR	45	Redundant test	
	46	Saknat test	
	48	Organisation	
	50	Respons	Förväntat resultat saknas/felaktigt i testinstruktion
	52	Testförutsättningar	Systemläge vid testfallets start saknas/felaktigt
	53	Testavslutning	Systemläge vid testfallets slut saknas/felaktig
STLDD &	61	Struktur	Indelning i moduler/delar olämplig
SDDD	62	Gränssnitt	
	63	Namngivning	olämplig/[följer ej regler]
	71	Realtidsproblem	kapplöpning/dödläge/handskakning/synkronisering
	72	Logiskt fel	
	73	Datafel	
	74	Timerproblem	
	75	Effektivitetsproblem	
Övrigt	100	Övrigt	

How to collect metrics?

- Everyone must contribute
- Procedures necessary
- Tool for metrics collection (and change management) from department available (ePUSS)
  - But other solutions possible (as long as they are clearly described in your SDP
- Forms used at seminar (+ available for illustrative purpose)

Changes to the process?

- Allowed, as long as it is clearly described in the SDP
  - And approved at review 1
- (However, it requires a lot of effort to change too much…)

Evaluation criteria

- for the project
  - process
  - product
  - final report
- for the individual
  - individual report

The process

- how well do you follow the specified process?
  - measured through document library and ePUSS (or similar)
  - checked mainly on three occasions (SSR, PDR and PR)

The product

- what quality does the product have?
  - product level
    how limited/simplified is the product?
  - specification compliance
    how does the delivered product comply with the SRS?
  - reliability and robustness
    how well have you managed to assure the quality of the product?

Final report

- extent
  - how far does the report get in:
    WHAT has happened (data reporting)
    WHY it happened so (cause analysis)
    WHAT can be done to improve (SPI)
- quality
  - how well are the chosen topics reported

Individual report

See course homepage from Friday next week.

Start thinking about it now, and start ”collecting data”

Objectives:

- to stimulate reflection on large-scale software development continuously through the course,
- to encourage a viewpoint on industrial practice in large-scale software development,
- to build on and integrate with what you have learned from previous courses,
- refering to research literature.

Uträkning av slutbetyg

Område	Viktning område	Del	Viktning del
Process	30%	Granskn 1	25%
		Granskn 2	25%
		Slutinlämning	50%
Produkt	30%	Granskn 1	20%
		Granskn 2	30%
		Slutinlämning	50%
Slutrapport	20%	Nivå	50%
		Kvalitet	50%
Individuell rapport	20%	Nivå	50%
		Kvalitet	50%

Värde	Förklaring
0	Underkänd. Kompletterande uppgifter krävs.
0,25	Underkänd, men resultatet kan kompenseras av andra moment.
0,5	Godkänd
0,75	Väl godkänd
1	Mycket väl godkänd

Poäng	slutbetyg
0 - 0,49	Underkänd
0,5 - 0,69	3
0,7 - 0,84	4
0,85 - 1	5

To do before exercise session 1

- book your exercise times
- study carefully the course programme
- familiarize yourself with the course material
- read PH chap. 1 through 4.4 (before exercise 1)
- get an overview of the development model (PH:A)
- Check out the course home page:
  http://cs.lth.se/ETSN05

Before your first project meeting

- check email often (project managers call for meeting)
- study carefully the tasks of all roles (PH:3)