

IN5140 – Smart processes and agile methods in software engineering

Lecture 23 August 2023 – part 2:

Course overview, Challenges in system development, System development processes

Yngve Lindsjörn

E-mail: ynglin@ifi.uio.no

Antonio Martini

E-mail : antonima@ifi.uio.no



Structure

- Part 1: Course information
- Part 2: Software development and challenges
- Part 3: Software processes

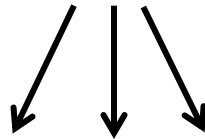


Topic: Agile, Lean and Process Improvement

Whole organization: Lean processes



System development: Agile/lean processes, methods and techniques



Scrum ... Kanban

Process improvement

Software Process Improvement

Learning Outcomes

At the end of the course you will ...

- have gained a good understanding of modern software development processes, including Lean and Agile methods
- know characteristics and effects of different development processes (process models)
- be able to contribute to efficient organization, conduct and improvement of software development processes



Student Project

- Prepare a (realistic) software process *improvement* (SPI) plan for a software development organization
- What could be improved?
 - a process,
 - an activity of a process or
 - a method/technique used in an activity or a process
 - ...



Project – Topic Ideas

Examples of problems and related improvement goals:

- *Customers find too many defects* – Improve software reliability
- *Inaccurate planning / estimates* – Improve planning methods/models
- *New technologies or standards make their way into the market (e.g., TDD, test automation, cloud computing..)* – Adapt existing processes to the new technologies and standards.
- *Software is hard to maintain / difficult to evolve* – Improve software architecture and associated processes
- *Increasing competition* – Speed-up development, issue releases more frequently
- *Customer are dissatisfied with deliveries* – More customer participation and more flexible process
- *“Old-fashioned”, heavy development process* – Modernize development processes, methods, and tools
- *Little diffusion of competence, low motivation* – Improve training & enhance involvement of people



Project Case – System/Development Organization

- System/software development organization: real or fictional
- Suggested improvement actions must be clearly related to identified problems and defined goals
- Find a realistic approach to solving a realistic problem, challenge or issue. Use your imagination

Project Case

- Contact a software development organization
 - Not necessary to mention the organization's name
 - If you find (or is involved in) a real-world improvement project, don't make yourself completely dependent on the reality, because a real-world project might have a longer time-frame than IN5140
- Alternatively, select one of the cases at the subsequent slides:



Case 1: Improving team communication in a distributed project

Problem:

An organization is working on a project regarding a software system, which is their main product. There are in total 10 teams working on the software, with 6-9 persons on each team. The teams are autonomous/self-organizing. The project is distributed, and of the 10 teams, 4 of them are located in India, 6 in Norway. During the Corona-situation, the teams had to work from home. After coming back, some team members have continued working from home, while others began working from the office. This has made the teams themselves distributed. The management of the organization feel like the teams are working in different directions, without a common goal. This has resulted in project deliveries often being delayed. They have also received feedback that the teams find it difficult to co-operate for several reasons, including using different working methods, documentation methods and other aspects of their processes. The management is worried that the communication difficulties will affect the quality of the product in the long run.

Description of organization:

The roles involved in the project mainly consist of managers, developers, testers and scrum masters. The organization uses Scrum as their process model, with emphasis on the teams being agile and self-organizing.

How the problem was identified:

Management realized that there was a large delay between the estimated and released product. When investigating the problem, they were told that the teams had too much to do and were therefore not able to meet the deadlines. Management decided to investigate further and found that the teams were doing a large amount of unnecessary, overlapping work. The teams had little to no contact in the beginning of the sprints, but suddenly had to familiarize themselves with large amounts of other team's code and tests at the end, and thus had to adjust their own work. This caused not only delays, but also stress on the employees.

Goal:

Reduce the delays and other waste by improving communication both between and within the project teams. The teams should work towards a common goal, while minimizing time spent on overlapping work.

Current process:

The process for the teams resembles Scrum. Every sprint begins with the team having a sprint planning where they go through the backlog. Then, development and testing starts. In the last week of the sprint, a sync meeting is required between the teams. The meeting often results in adjustments needed to be made to the code, new tests, etc. If the team is not able to finish the tasks in their backlog, those tasks are moved to the next sprint.

Case 2: Improving the efficiency of software maintenance

Problem:

An organization has a large number of systems that are continuously maintained. Maintenance is defined as the work involved in user support, handling change requests, error correction and small functional enhancements and the deployment of new versions of the systems to its users. The organization experiences that there is a growing back log of change requests from the users and feels that the current maintenance process is too time consuming.

Description of organization

Approximately 50 people work on maintaining 6 different systems. They are organized in 2 groups, each with of 3 teams. One group handles 3 systems, and the other group handles the other 3 systems. Each group has a manager and each team has a team lead. The teams are 1. User support who handles questions and requests from the users of the systems, 2. Requirements and test with functional specialists who specify new requirements for the system and subsequently test that these are fulfilled, and 3. Development with developers who correct errors and implement the required changes.

How problem was identified

The organization was compared to other similar organizations in a benchmarking, and a result of the benchmarking was that it became clear that this organization spends a lot more effort than other similar organizations on software maintenance. In particular, much effort is spent on communication between the teams about the change requests and on sending these back and forth between the teams. Furthermore many defects slip through testing.

Goal:

The goal is a more efficient maintenance process where less time is spent on communication and defects are detected earlier.

Current process

The current process is very similar to the V-model.

Current roles, activities, artifacts and tools

The change requests, called CRQs, are registered by User support in a system called Repair. Each CRQ contains a description of the problem and a suggested solution in the form of new requirements. When Development has implemented the requirements they tick off a box for development completed. Correspondingly a box for test completed is ticked off by Requirements and test when they have completed testing.



Case 3: Improving requirements handling for a software product line

Problem:

A company maintains software for a product line of access control systems. When software is to be developed for a new kind of lock, a new project is set up. The company has no formalized process for requirements handling, so each new project tends to define their own way of documenting and handling requirements. This also means that requirements are not transferred in a systematic way when the system is set in production and the software is transferred to maintenance. Many requirements are similar across projects, but due to the lack of a formalized process for handling requirements the reuse of requirements from one project to another depends on the people who are involved. The company has recently experienced tougher competition and is looking at ways to become more competitive.

Description of organization

The company primarily consists of project managers, developers and product specialists, organized in projects. In addition some people are employed in administrative functions.

How problem was identified

The company recently hired a new employee who told the management that in his former company requirements management was done in a more formalized way and that that increased their competitiveness.

Goal:

A formalized process for requirements handling that reduces time-to-market for new software products, and in particular reduces the necessary effort to start a projects.

Current process

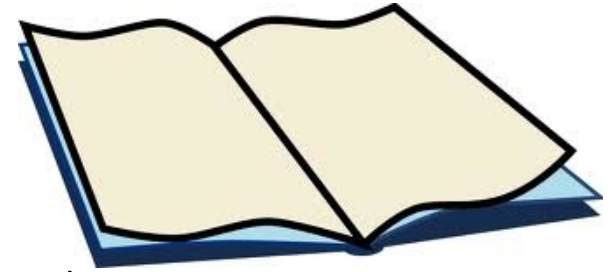
Scrum is used in development, but scrum relies on a back-log of requirements, for example in the form of user stories, and does not incorporate requirements handling.

Current roles, activities, artifacts and tools

Each project has a project back-log in the tool Jira with user stories and tasks to be performed in the project.



Project: Report Structure



- Cover page (Title, Authors, Date, E-mail, Course ID, ...)
- Table of contents
- Contents:
 - *Improvement context* (type of company, product, process) and improvement goal(s) (what? how much? when?)
 - *BPMN model* of the existing process that you will improve (Deliverable 1)
 - *Suggested changes* of SW development process (What? / Old process vs. New process)
 - Implementation of process changes (When? Who is responsible?)
 - *Monitoring/Control* (Identify and describe measures to be used to assess effects of process changes. Based on the improvement goal(s) in your project, identify and describe minimum three measures to be used to assess effects of process changes. For each measure, describe: Who will collect/report data? When (how often) will data be collected? How is data collected, for example using which tools? How is data quality and validity ensured, who is for example responsible? Also discuss possible challenges related to data collection and data validity. Note that you do not have to actually collect all data for all the measures if practically difficult.)
 - *Discussion* (Why? Improvement methods applied, rationale for changes, risks)
 - *References*

Project: Groups

- Group submissions, groups of three to five students
 - You are allowed to form groups yourself. But we will also form groups if needed. Forming groups will start in the seminar groups next week (group teachers: Karthik Shivashankar (karths@ifi.uio.no), Gyda Sæther (gydaes@ifi.uio.no),
- Expectations
 - Length: 20-25 pages from 3 persons, and 22-27 pages from 4-5 persons



Project: Dates

- 21 Sep: Deliverable 1 – Process Model
- Oct (dates will be added later)
Project Presentations (5 min)
- 09 Nov: Deliverable 2 – Project Report
- Deliverables should be submitted as a pdf-file in Devilry before 24:00 on the submission day
- 04 Dec at 15.00 (4 hours): written, *digital* exam (Inspira), at Silurveien



Course Evaluation, Marking and Grades

Part 1: Project / 30% of grade [30 marks]

- Deliverable 1: Process modeling [approved/not approved]
- Short presentation [compulsory]
All students should contribute in the presentation.
- Deliverable 2: Full project report [30 marks for material not included in Deliverable 1]

Evaluation criteria for the Project Report:

- Content
- Clarity/Conciseness
- Formality (cover page, captions, referencing, etc.)

Part 2: Written exam / 70% of grade [70 marks]

Subject:

Questions about the course topics (lectures, materials)

Evaluation criteria:

- Correctness and completeness
- Clarity and conciseness
- Relevance (→ is the answer to the point?)

Total marks in the course: 100



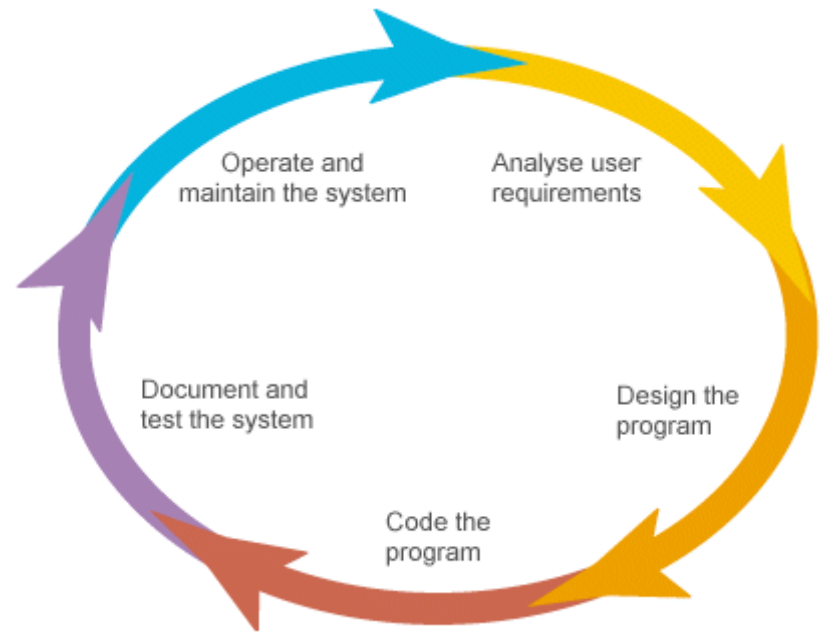
Structure

- Part 1: Course information
- Part 2: Software development and challenges
- Part 3: Software processes



Software Development

- Overall topic of IN5140:
How to develop software systems?
 - Software development includes all activities involved between the conception of the desired software and the operationalization of it
 - May include new development, prototyping, modification, reuse, maintenance, or any other activities that result in software products



Software Maintenance = All Changes Made to a System after it has Become Operational (Put into Use)

- Types of software maintenance
 - Fixing bugs (**corrective maintenance**)
 - Improving structure (re-engineering, refactoring) (**preventive maintenance**)
 - Adapt to changing environments (**adaptive maintenance**)
 - New functionality (**perfective maintenance**)
- Constitutes most of the costs in the total lifetime of a system
- Sometimes implicit in the definition of software development, sometimes not



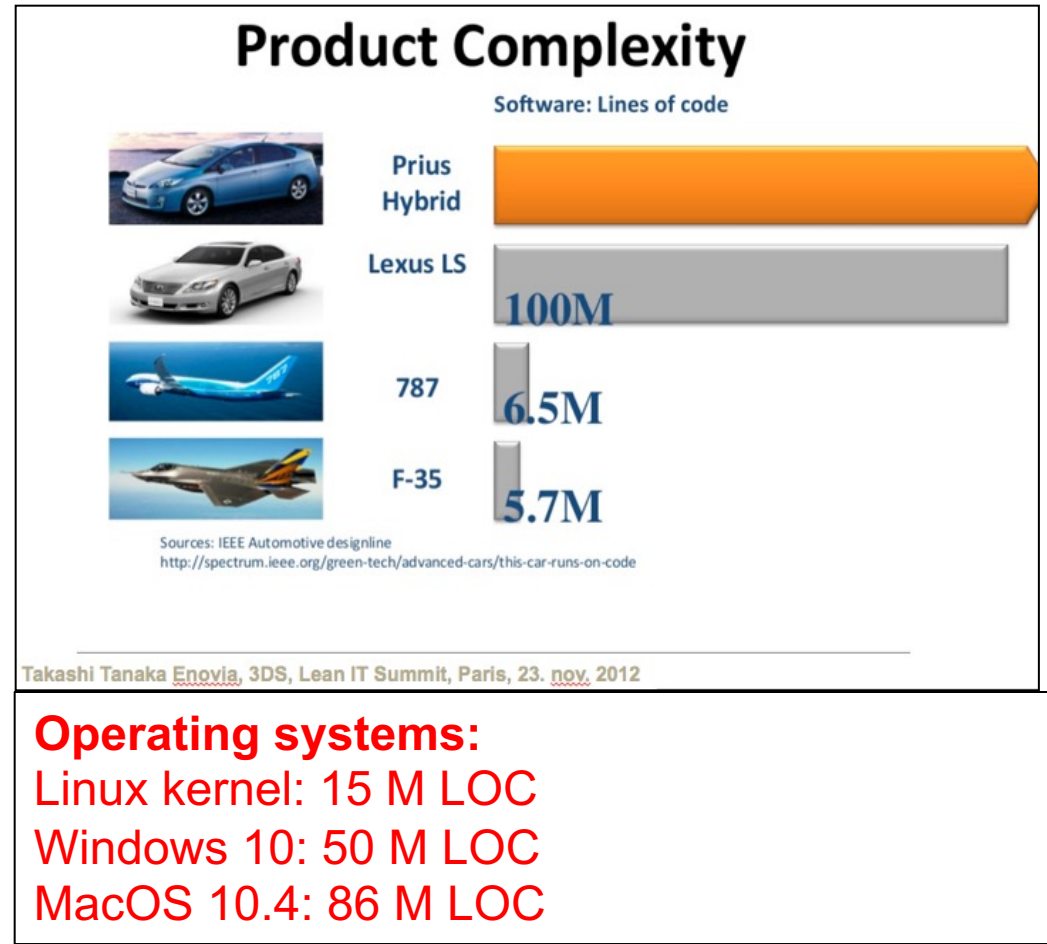
Software is widespread and used in

- **industry**
 - transportation, energy, telecom, medical, fishing, agriculture, etc.
- **administration**
 - banking, insurance, e-government, information systems, payroll systems, HR (human resource) systems, ERP (enterprise resource planning) systems, ...
- **science**
 - medicine, biology, chemistry, physics, ...
- **consumers**
 - entertainment, education, information, assistance, ...
- ...

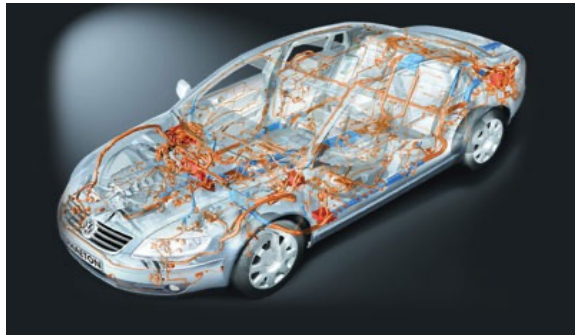


Variation in Size and Complexity

- Software
 - From a few thousands to millions lines of code (LOC), several hundred thousands pages
- Development teams
 - From 1 to more than 1000 developers (MS Windows 7)
- Costs (development and maintenance)
 - From a few thousands to billions



Example of a Software-Intensive System



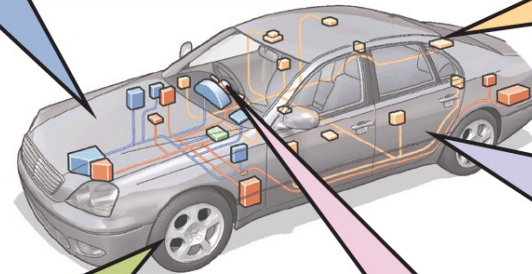
- Example: Embedded systems in a car

Engine and power train

Adaptive cruise control (ACC)
Electronic controlled transmission (ECT)
Electronic fuel injection (EFI)
Hybrid vehicle control (HVC)

Comfort and pleasure

Back guide monitor
Climate control
Door lock control
Immobilizer
Night view
Power seat, power windows
Remote engine start (keyless)



Communication

Data communication module (DCM)
In-vehicle network: CAN, LIN, MOST,
FlexRay, TTEthernet
Short-range wireless: Bluetooth

Chassis and safety

Adaptive front-lighting system (AFS)
Active control suspension (ACS)
Antilock brake system (ABS)
Electronic stability control (ESC)
Pre-crash safety (PCS)
Traction control (TRC)
Vehicle dynamics management (VDM)

Human interface and support

Digital instrument panel (LCD)
Head-up display (HUD)
Car navigation system
Speech recognition system
Touch panel, haptic interface

Source:
Mikio Aoyama: Computing for
the Next-Generation
Automobile, IEEE Computer,
pp. 32-37, June, 2012

What's Special about Software?

- Software is developed, not produced – every software product is unique
 - Software development
 - is a design process,
 - is often experimental, and
 - requires knowledge, skills and creativity
 - Software is not tangible (not “physical”)
 - Natural laws irrelevant (only relevant for HW)
 - Often large design teams
 - High degree of flexibility/changeability of artefacts
 - Measurements of size, complexity and quality is not (yet?) well-defined (different to, for example, physics)
- } **“Soft Factors”
dominate!**

Challenges with Software/Systems

- Unsatisfactory quality (incl. functionality)
- Exceeding budget (cost/effort)
- Late delivery (lead time problems)

Source:
**"Extreme Chaos" by The
Standish Group (2001):**



Reasons for Software Project Failures

Table 3

**Reasons for project cancellation
with percentages and 95% confidence intervals
for the 2007 respondents ($n = 18$)***

Reason for cancellation	Percentage of respondents (95% confidence interval)
Senior management not sufficiently involved	33 (13, 59)
Too many requirements and scope changes	33 (13, 59)
Lack of necessary management skills	28 (10, 54)
Over budget	28 (10, 54)
Lack of necessary technical skills	22 (6, 48)
No more need for the system to be developed	22 (6, 48)
Over schedule	17 (4, 41)
Technology too new; didn't work as expected	17 (4, 41)
Insufficient staff	11 (1, 35)
Critical quality problems with software	11 (1, 35)
End users not sufficiently involved	6 (0, 27)

**The 95% confidence interval is usually wide because we're looking at only 18 cancelled projects. The respondents had the option of adding qualitative information as well as the predefined categories.*

K. El-Emam and A.G. Kouru.
A Replicated Survey of IT
Software Project Failures,
IEEE Software, 25(5):84-90,
2008.

SW Project Risks

(from 12 surveys)

Source:
 Top Ten Lists of Software Project Risks: Evidence from the Literature Survey by Tharwon Arnuphaptrairong, Proceedings of IMEC 2011, March 16-18, 2011, Hong Kong

TABLE VI
 TOP TEN LIST OF SCHMIDT ET AL. [6], (FINLAND)

Rank	Software Risk	Dimension
1	Lack of effective project management skill	P&C
2	Lack of top management commitment to the project	Org
3	Lack of required knowledge /skill in the project personnel	T
4	Not managing change properly	P&C
5	No planning or inadequate planning	P&C
6	Misunderstanding of the requirements	R
7	Artificial deadlines	P&C
8	Failure to gain user commitment	U
9	Lack of frozen requirements	R
10	Lack of "people skill" in project leadership	T

TABLE VI
 TOP TEN LIST OF SCHMIDT ET AL. [6], (USA)

Rank	Software Risk	Dimension
1	Lack of top management commitment to the project	Org
2	Misunderstanding of the requirements	R
3	Not managing change properly	P&C
4	Failure to gain user commitment	U
5	Lack of effective project management skill	P&C
6	Lack of adequate user involvement	U
7	Failure to manage end-user expectations	U
	Lack of effective project management methodology	P&C
	Unclear / misunderstood scope/ objectives	P&C
	Changing scope / objectives	P&C

TABLE XI
 SOFTWARE RISKS FREQUENCY BY DIMENSION

Dimension	Total frequency
User	14
Requirement	17
Complexity	4
Planning and Control	27
Team	9
Organizational Environment	9
Total	80




Goal of IN5140

Teach you to help avoid software project failures by

- identifying risks and
- means to tackle them



Structure

- Part 1: Course information
 - Part 2: Software development and challenges
 - Part 3: Software processes
- 

What is a Process?

- – a set of activities carried out to fulfil a job
- Usually wish to run a process that is carefully planned
- Everything we do involves processes, for example:
 - Plan a project
 - Prepare a meeting
 - Run a meeting
 - Write a requirements specification
 - Design a web-site
 - Program a function
 - Run a test

System Development Process

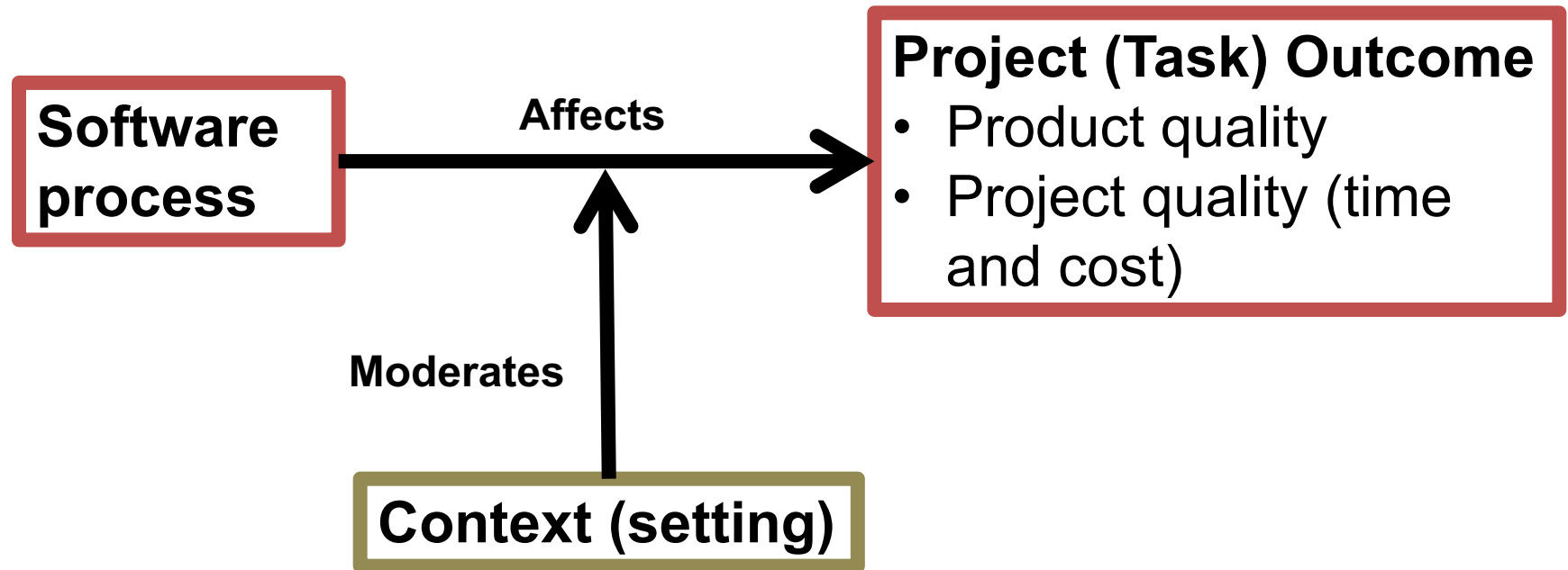
- System development process (= software process): the activities that are performed to develop a software system and the ***way they are performed***
- The activities vary but will always include the elements:
 - Specification of requirements, that is, what the system should do and the quality of how its done
 - the way requirements are specified may vary, which leads to different requirement specification processes
 - Design of the system
 - Coding (programming)
 - Validation that the system satisfies the needs of the customer or user
 - Evolution of the system, that is, changes according to new or changed requirements



Process Affects Properties of both Project and System being Developed

- The development process and the context in which it is performed affect the quality of project properties and the resulting system or product being developed
- The process also affects the work environment (work satisfaction, motivation, competence development, etc.), which in turn affects project and product quality

Process and Project (Task) Outcome



Aspects of Process

- Which activities are in focus in the process?
- How much effort is spent on each of them?
- A process may also include
 - Which and the way methods, techniques, tools and practices are used
 - Parts of the products/results of an activity
 - Roles of those involved in the process



Examples of Roles

- Developer
- Maintainer
- Architect/System designer
- Graphical designer
- Documenter
- Tester
- Project manager
- User/customer representative

Employing a project with the right competence is far from trivial!

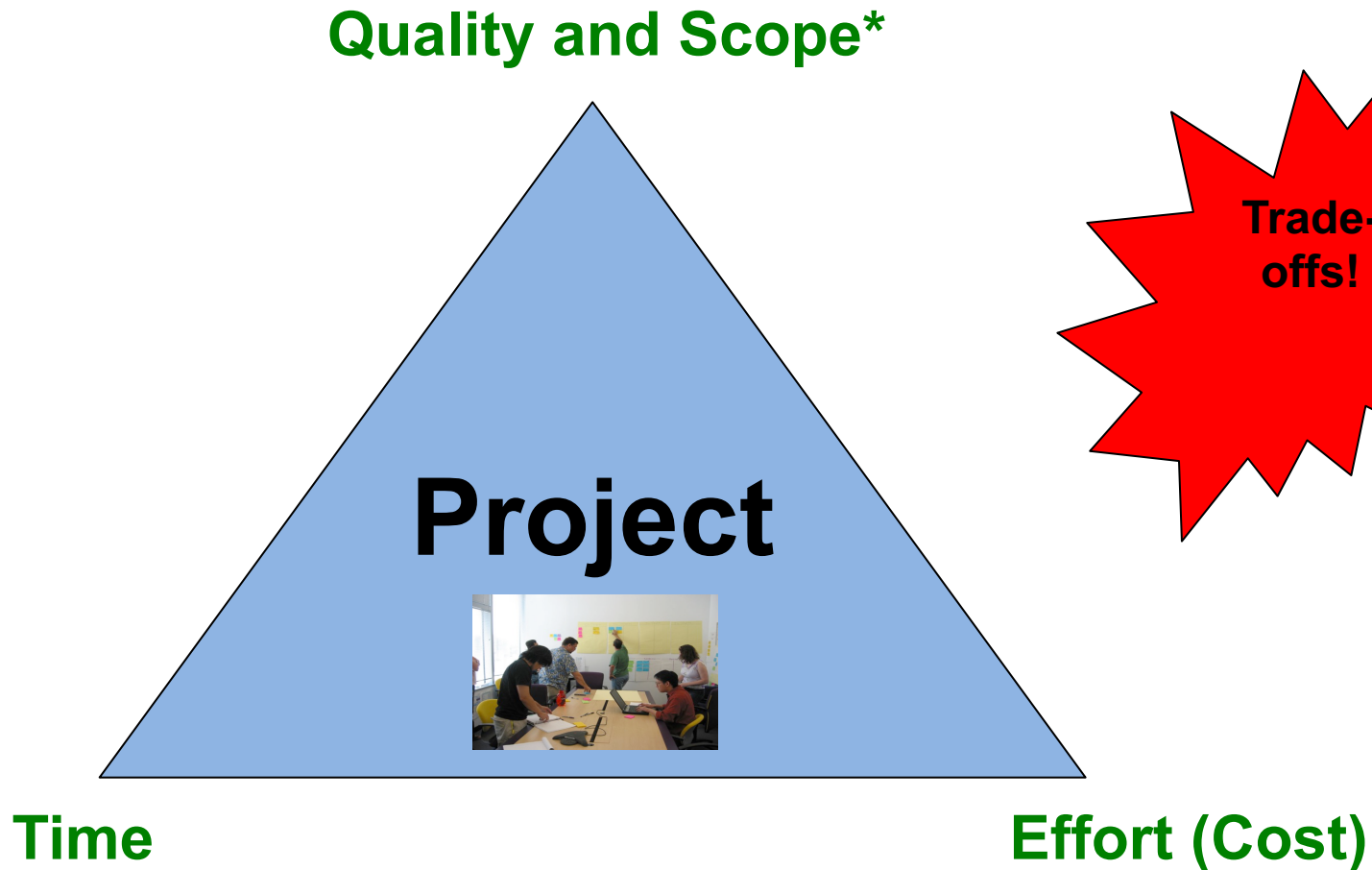
Examples of Tools

Tools for:

- Development (IDE)
- Modelling / Diagram construction
- Configuration management/version control
- Testing
- Bug & issue tracking
- Project management

Choice of tools is also not trivial

The Magic Triangle



*the extent of the functionality of the system

Before Next Lecture

- Get the book
- Start thinking about your project
 - In the first group lecture, you should indicate whether to create a group yourself instead of letting the group teacher do it

- Homework exercise

Define the concepts as follows and describe attributes and/or aspects of them:

- Software system quality
 - For each of these attributes, argue for what kind of stakeholders (“interessenter”) (users, owners, developers, etc.) would this attribute be relevant for?
- Software process
- Software process improvement
- Software project

