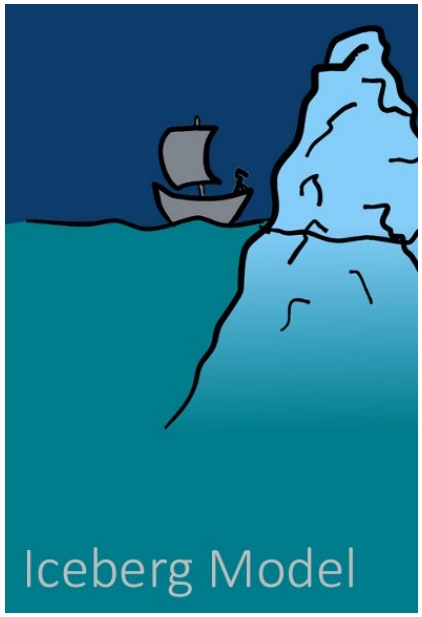
AGILE SOFTWARE ENGINEERING:

RISK MANAGEMENT

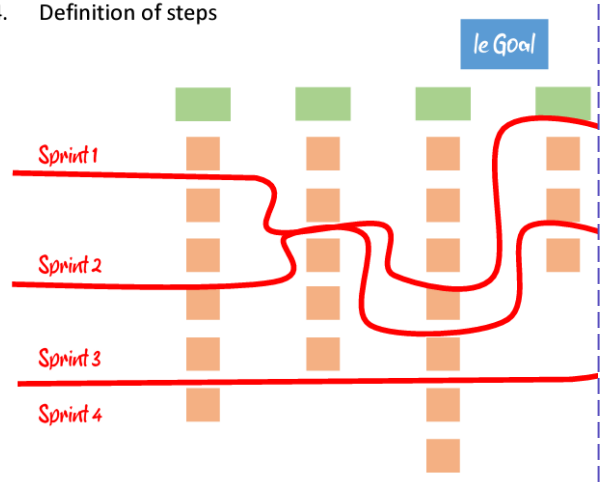
JOHN STOUBY PERSSON

AALBORG UNIVERSITY
DENMARK

Last lecture by Karsten Jahn



1. Goal definition
2. Brain storming
3. Grouping & ordering
4. Definition of steps



Lecture objectives

Knowledge about risks in software engineering

Skills in software risk identification, assessment, mitigation, monitoring, and management.

Competencies to manage risks in software engineering.

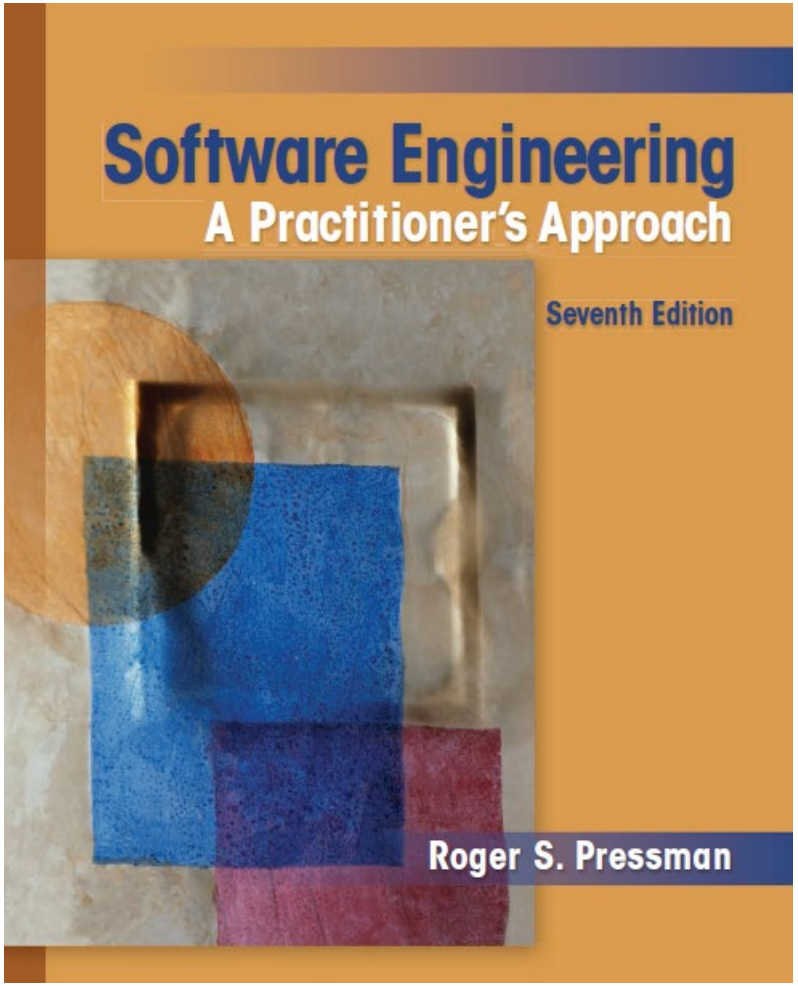
What is ‘risk’?

- Concerns the future
- “If you don’t actively attack the risks, they will actively attack you” --- Tom Gilb

- ”the possibility of loss or injury” --- Barry Boehm
- Proactive
  - Potential risks are identified, likelihood and severity assessed, prioritised
  - A risk mitigation plan addresses the identified risks

Software risk

- Project risks
- Technical risks
- Business risks
- Known risks, predictable risks, and unpredictable risks



Risk Management Activities

- Risk assessment
  - Risk identification
  - Risk analysis
  - Risk prioritisation
- Risk control
  - Risk reduction
  - Risk management
  - Risk resolution

Risk Identification

- Systematically identify known and predictable risks to the software project
- Generic risks potentially threaten all software projects
- Product-specific risks are specific to the technology, the people, the project's context
- Risk frameworks are useful

Identification: Boehm's risk item list

Risk Item	Risk management techniques
1. Personnel shortfalls	Staffing with top talent, job matching; teambuilding; morale building; cross-training; pre-scheduling key people
2. Unrealistic schedules and budgets	Detailed, multisource cost and schedule estimation; design to cost; incremental development; software reuse; requirements scrubbing
3. Developing the wrong software functions	Organization analysis; mission analysis; ops-concept formulation; user surveys; prototyping; early users' manuals
4. Developing the wrong user interface	Task analysis; prototyping; scenarios; user characterization (functionality, style, workload)
5. Gold plating	Requirements scrubbing; prototyping; cost–benefit analysis; design to cost
6. Continuing stream of requirement changes	High change threshold; information hiding; incremental development (defer changes to later increments)
7. Shortfalls in externally furnished components	Benchmarking; inspections; reference checking; compatibility analysis
8. Shortfalls in externally performed tasks	Reference checking; pre-award audits; award-fee contracts; competitive design or prototyping; teambuilding
9. Real-time performance shortfalls	Simulation; benchmarking; modeling; prototyping; instrumentation; tuning
10. Straining computer-science capabilities	Technical analysis; cost–benefit analysis; prototyping; reference checking

Boehm, B. W. (1991). Software risk management: principles and practices. *IEEE software*, *8*(1), 32-41.

Identification:
Ranked risks

1. Commitment from top managers (software and customer)
2. Commitment from end-users
3. Requirements fully understood
4. Involvement of customers
5. Realistic end-user expectations
6. ...

Ranked based on a large study with many respondents across many companies

Identification:
Risk categories

- Product size (PS)
- Business impact (BI)
- Customer characteristics (CCh)
- Customer communication (CCo)
- Tools (To)
- Technologies (Te)
- Staff size and experience (SSE)
- …

Risk analysis

- For all risks:
  - Estimate probability
  - Estimate loss (impact)
  - Calculate exposure
  - Describe consequences

$$\text{Exposure}(r) = P(r) \times L(r)$$

Risk analysis

Risk	P(r)	L(r)	E(r)	Consequence
Inexperienced staff	0.6	9	5.4	Less quality Less control More overhead
End user resistance	0.4	4	1.6	No participation in prototyping -> no feedback
Less resources than planned	0.7	7	4.9	Delays Surprise
Poor testing	0.1	4	0.4	Does not demo
…	..	..	..	…

Risk prioritisation

Risk	P(r)	L(r)	E(r)	Consequence
Inexperienced staff	0.6	9	5.4	Less quality Less control More overhead
Less resources than planned	0.7	7	4.9	Delays Surprise
End user resistance	0.4	4	1.6	No participation in prototyping -> no feedback
…	..	..	..	…

- ☐ Sort according to exposure
- ☐ Establish cut-line

Risk management planning

- Establish the plan to reduce and resolve the risks
- Sometimes referred to the RMMM Plan (Risk Mitigation, Monitoring, and Management Plan)

Risk	P(r)	L(r)	E(r)	Consequence	Mitigation plan
Inexperien-ced staff	0.6	9	5.4	Less quality Less control More overhead	Meet w/staff to assess skill level Plan on-the-job training Specialise staff Pair programming
Less resources than planned	0.7	7	4.9	Delays Surprise	…
End user resistance	0.4	4	1.6	No participation in prototyping -> no feedback	…
…	..	..	..	…	

Risk resolution

- Perform the planned risk management activities to reduce and monitor risks

Risk	P(r)	L(r)	E(r)	Consequence	Mitigation plan	Monitor
Inexperienced staff	0.6	9	5.4	Less quality Less control More overhead	Meet w/staff to assess skill level Plan on-the-job training Specialise staff Pair programming	Measure quality of work Discuss w/staff
Less resources than planned	0.7	7	4.9	Delays Surprise	…	…
End user resistance	0.4	4	1.6	No participation in prototyping -> no feedback	…	…
…	..	..	..	…		

Risk resolution

- Develop contingency plan

Risk	P(r)	L(r)	E(r)	Consequence	Mitigation	Monitor	Management
In-experience	0.6	9	5.4	Less quality	Meet w/staff to assess skill level Plan on-the-job training Specialise staff Pair programming	Measure quality of work Discuss w/staff	Bring in experienced staff
	0.7	7	4.9	Delays Surprise	…	…	
	0.4	4	1.6		…	…	
…	..	..	..	…			

Now your Risk Mitigate Monitor Management (RMMM) plan is ready

AALBORG UNIVERSITY
DENMARK

Agile risk management

Disagreement about whether risk management is needed, e.g. all roles and events in Scrum can be argued to reduce risk.

Roles in Scrum

- PO works fulltime to reduce risks related to product and stakeholder
- SM and team works fulltime to reduce risks from the organization or technical risks

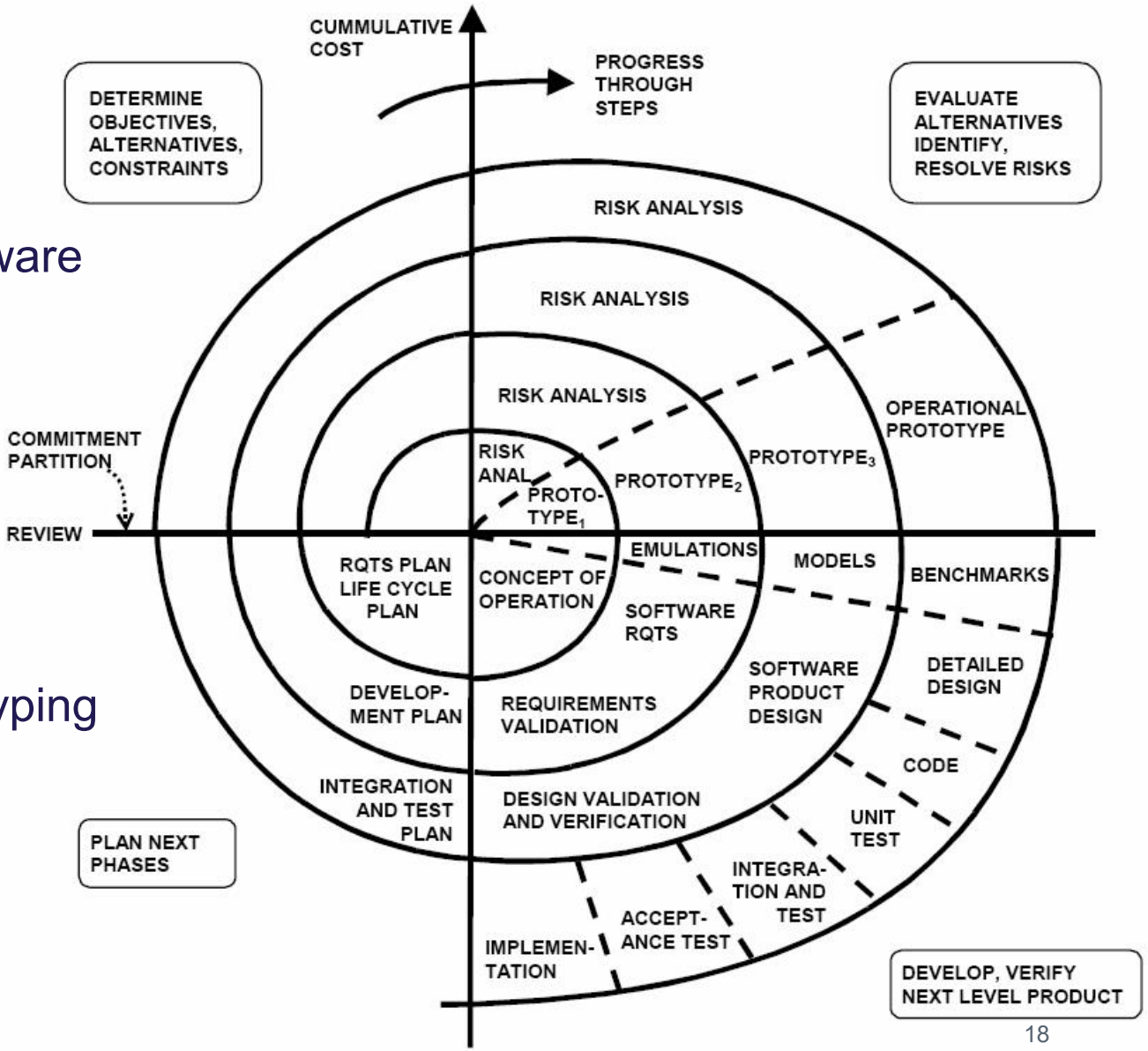
Meetings in Scrum

- Sprint Planning – alignment with PO, and slicing of tasks are ways to reduce the risk
- Daily Scrum – Any impediments? Ensures a team conversation that could lead to risk mitigation
- Sprint Review and Sprint Retrospective foster conversations to reduce risks in product (review) and ways of working (retrospective)

Risk analysis for development process decisions

Spiral model for software development
(Boehm 1988)

Specifying vs. prototyping
(Boehm et al. 1984)

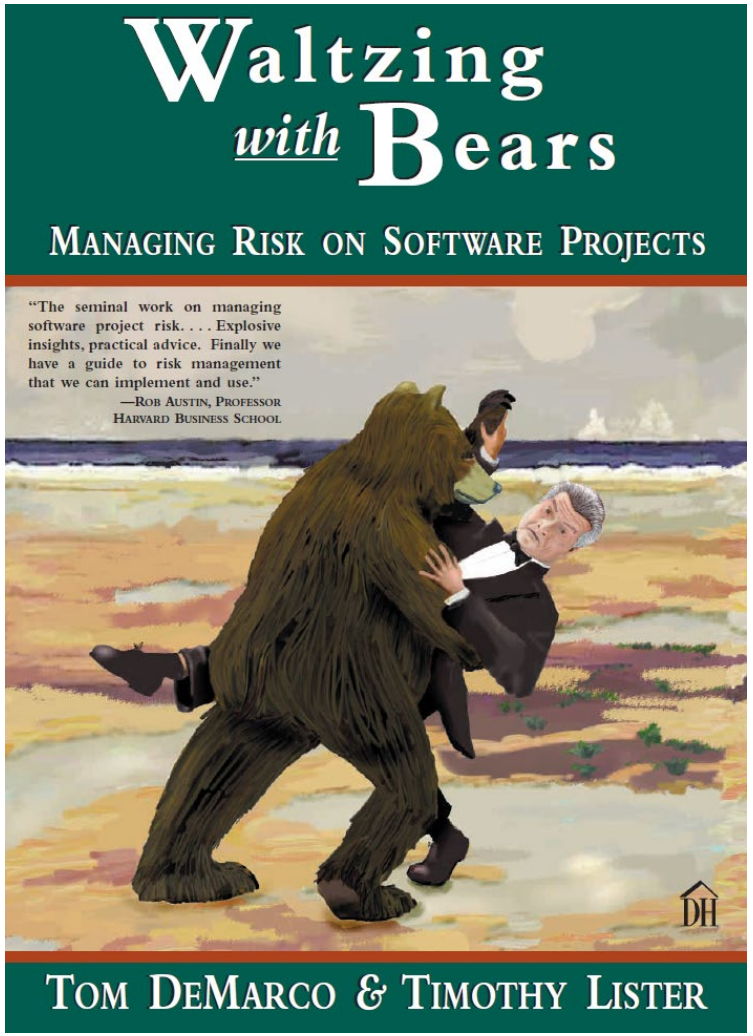


Incrementalism for risk mitigation

"The best bang-per-buck risk mitigation strategy we know is incremental delivery"

--- DeMarco & Lister 2003

Ethics of belief: Risk management helps to act ethically in arguing your **right** to believe in success



Group exercises

Exercise 1: Identify at least 15 risks for your semester project or the imaginary product. (30%)

Exercise 2: Assess the impact and probability for each of the above risks on a 9 point scale (high/medium/low) to make a list sorted by risk exposure = Impact (1-9) x probability (1-9). (20%)

Exercise 3: Use the table below to make a plan for the five risks with highest risk exposure, including details on their consequence, mitigation plan, monitoring, and management contingency – RMMM. (50%)

Risk	P(r)	L(r)	E(r)	Consequence	Mitigation plan	Monitor	Management Contingency
#1							
#2							
#3							

Lecture objectives

Knowledge about risks in software engineering

Skills in software risk identification, assessment, mitigation, monitoring, and management.

Competencies to manage risks in software engineering.