



SOFTWARE ENGINEERING LECTURE 6

FALL 2018
CARSTEN RUSENG JAKOBSEN



AALBORG UNIVERSITY
DENMARK

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

- Uncertainty
- Loss
- 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



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	10	6	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	10	6	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
Inexperienced staff	0.6	10	6	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	10	6	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 Prevent	Monitor	Management Contingency
...	0.6	10	6	...	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



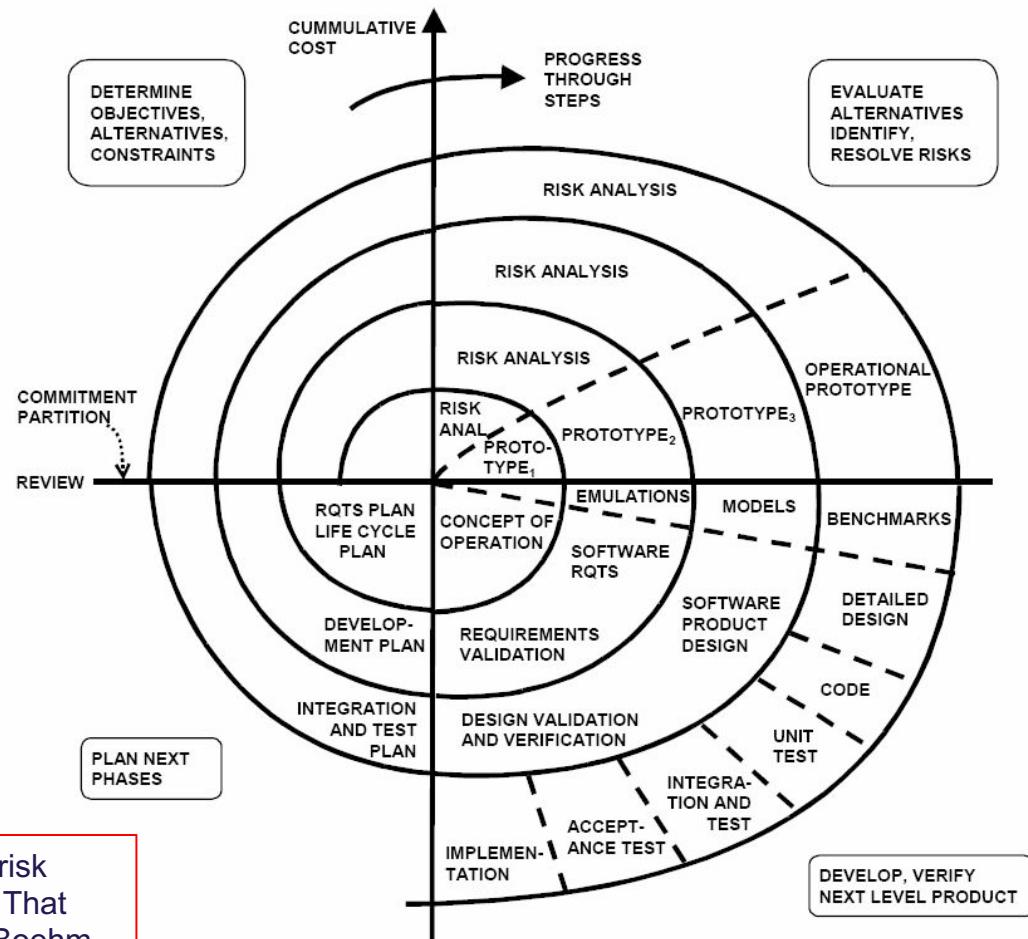
Agile risk management

- Disagreement about whether risk management is needed
- Control by addressing risks
 - where to start (reduce risks first)
 - progress
 - e.g., in the incremental process model
- Spiral model – a mixed approach



Spiral model

- Mixed method
- Spiral model for software development [Boehm 1988]
- Controlled by risk management



Let me take you back a bit in history to where risk management came into software engineering. That happened as early as in the mid-80ies where Boehm used it in his spiral model and in making his claims about when to specify (follow a traditional process) and prototyping (follow an agile process).

