



# SOFTWARE ENGINEERING CURRICULUM

CARSTEN RUSENG JAKOBSEN



AALBORG UNIVERSITY  
DENMARK

# Exam procedure

- The topic will be chosen at random when the exam begins. The student will then be expected to give a five minutes presentation showing a detailed knowledge of the chosen topic. The exam will then develop as a 12-minutes conversation in which the student will be expected to show a detailed knowledge of any or all parts of the readings, and relate that knowledge to their own understanding of and experience with software engineering.

Examinations will be conducted in Danish or English at the student's choice. Notes may be used during the examination to support recollection, but reading directly from the notes is not acceptable.

- The grade will be awarded for the candidate's performance at the oral examination.
- Reflective, evaluative and comparative answers will weigh higher than simple repetition of theory from the readings.
- It is a requirement to enter the exam that the mini project report has been submitted on time.



# Readings

- Ian Sommerville, Software Engineering, 10th edition, Pearson, 2015. Chapter 1.1-1.2, 2, 4.3
- Ian Sommerville, Software Engineering, 9<sup>th</sup> edition, Pearson. Chapters 8, 23, 25
- Pressman, chapter 28
- Larman, C., (2004). Agile & Iterative Development: A manager's guide. Addison-Wesley, Boston. Chapter 8-9.
- Schwaber & Sutherland: The Scrum Guide
- Dahlbom & Mathiassen, chapter 7.
- Tom De Marco & Timothy Lister - Waltzing with Bears (Prologue: Ethics of Beliefs)
- Humphrey 1997, chapter 25
- Pries-Heje et al. 2008. The Road to High Maturity
- Boehm, Barry, and Richard Turner. "Observations on balancing discipline and agility." Agile Development Conference, 2003. ADC 2003. Proceedings of the. IEEE, 2003.
- Weinberg's Software Engineering Cultural Patterns from Quality Software Management, Vol. 2, First-Order Measurement, Dorset House Publishing1993.
- Boehm: A spiral model of software development and enhancement
- Dyba, Tore, et al. "Are two heads better than one? On the effectiveness of pair programming." Software, IEEE 24.6 (2007): 12-15.
- ~~Nielsen, Peter A., One page hand out.~~
- All slides.
- All videos by Sommerville referred to in Moodle.



# Videos in curriculum

- Lecture 1: Plandriven and agile
  - Sommerville's video introduction to chapter 2 in his book
- Lecture 1: Ethics
  - Martin Fowler - Not just Code Monkeys (Ethics) (From 10m25s- 15m25s into the video)
- Lecture 5: Project Planning
  - Agile Planning with story points (11min)
  - Planning with Gant charts (3min)
- Lecture 7+8: Quality and User Involvement
  - Technical Review
  - Nordstrom Innovation Lab: Sunglasses iPad App case study



# Final exam topics – numbered list

1. Software process models: waterfall
2. Software process model: incremental and iterative
3. Software process model: integration & configuration
4. Comparison of plan-driven and agile software engineering processes, including analysis of home grounds
5. Key features of Scrum
6. Key features of RUP
7. Requirements Elicitation and Management
8. Managing change to requirements
9. Quality Control: Verification and Validation
10. Risk Management
11. Project Planning and Management
12. Quality Management: How is quality defined - agile versus plan driven approaches
13. Configuration Management

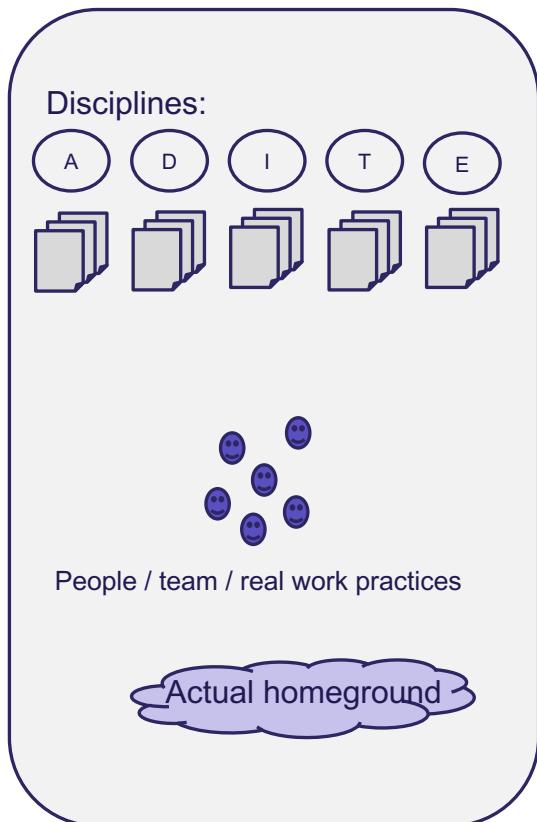


# SUGGESTION FOR GENERIC AGENDA ON EXAM QUESTION

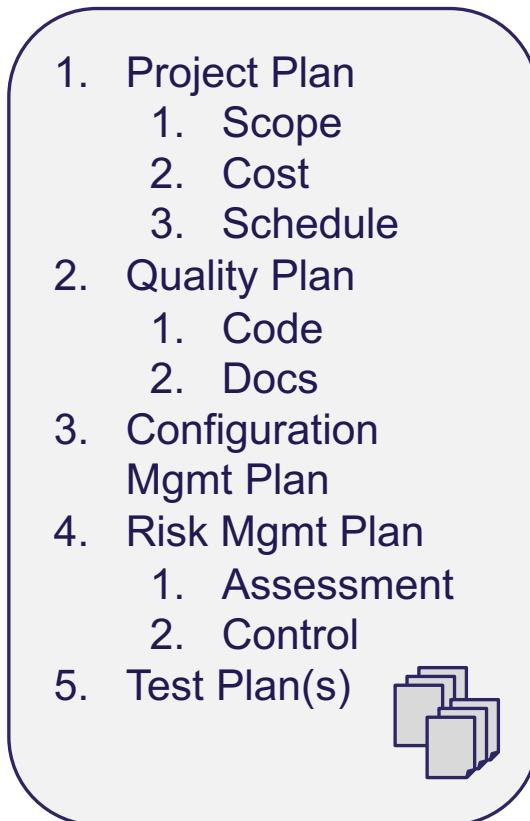
1. Define central concepts within question
2. What problems are addressed by the question
  1. Advantages
  2. Disadvantages
3. What other topics relate to this question



# SOE in one slide



Activities and work products



Plans

Proces models

- Waterfall
  - Inkremental / Iterative
  - Integration & konfiguration
- 
- Spiral model

Homeground analysis

Proces Improvement  
- Model e.g. CMMI  
- Metrics & experience

Procesmodels

# Q1: Software Process Model - waterfall

## Example on what questions could be

1. What is Software Engineering (SE) a response to? (complexity, failure)
2. What are SE process activities?
  1. (specification, design, development, validation, evolution)
3. What is a software process model (Set of related activities that leads to a software product)
4. Name the 3 typical process models: waterfall, incremental/iterative, integration and configuration
5. Describe characteristics of the waterfall model
  1. Activities in sequence, handover of workproducts between phases, milestones and related work products are used to monitor progress
6. When should you consider to use waterfall
  1. On projects: a) using embedded systems b) are life critical or c) very large
7. How can I decide if agile or waterfall is best fit for my situation? (Boehm: Analyse af home ground)
8. Plan driven versus agile processes?
  1. Describe them (activities in sequence versus all activities at the same time)
9. Describe how the incremental model works
  1. Can it be plandriven, can it be iterative? You can iterate within increments, you can have increments planned
10. Describe Integration and Configuration process model
  1. What risks are reduced (development risk is reduced by reuse – instead lack of owner ship of what is reused, introduces risk of not being able to make desired changes to reused parts, at all or in the desired time frame)



# Q2: Software Process Model – Inkrementiel / Iterativ

## Example on what questions could be

1. What is Software Engineering (SE) a response to? (complexity, failure)
2. What are SE process activities?
  1. (specification, design, development, validation, evolution)
3. What is a software proces model (Set of related activities that leads to a software product)
4. What are the 3 typical models: (Waterfall, Incremental/iterative, Integration and configuration )
5. Describe the incremental/iterative model
  1. You slice the big plan into smaller slices
6. Can it be both plandriven and agile?
  1. Yes, fx In plandriven you cut a preditive waterfall plan into slices, In iterative you work in fixed timeboxes and regularly update the full backlog for the project
7. What are the advantages of incremental/iterative model?
  1. A) price on requirements changes are less b) easier to get feedback c) customer gets earlier an opportunity to use part of the product and obtain the related value where use nd value comes in the very end when using waterfall
8. What disadvantages are there?
  1. A) The proces is invisible – management support for measurable progress can increase documentation cost b) a software systems infrastructure tends to deteriorate, as new increments are added
9. How can I determine if incremental/iterative or waterfall fits me? (Boehm: Analyse af home ground)
10. Plan driven versus agile processes?
  1. Describe them (activities in sequence versus all activities at the same time)
11. Describe how the incremental model works
  1. Can it be plandriven, can it be iterative? You can iterate within increments, you can have increments planned
12. Describe Integration and Configuration process model
  1. What risks are reduced (development risk is reduced by reuse – instead lack of owner ship of what is reused, introduces risk of not being able to make desired changes to reused parts, at all or in the desired time frame)



# Q3: Software Process Model – Integration and configuration. Example on what questions could be

1. What is Software Engineering (SE) a response to? Complexity, failure
2. What are SE process activities?
  1. (specification, design, development, validation, evolution)
3. What is a process model (Set of related activities that leads to a software product)
4. What are the 3 typical models: (Waterfall, Incremental/iterative, Integration and configuration )
5. Describe integration and configuration (software reuse) model
6. What advantages and disadvantages does it have?
  1. Advantage: Reduces amount of software to develop, whereby cost and risk is reduced.
  2. Disadvantage: Requirements compromises are inevitable and can lead to a system not meeting the real needs of the users
  3. Disadvantage: Normally you have no control of software you reuse, for instance how and when new versions are introduced and how the functionality is updated or changed
7. How can I determine if incremental/iterative or waterfall fits me? (Boehm: Analyse af home ground)
8. Plan driven versus agile processes?
  1. Describe them (activities in sequence versus all activities at the same time)
9. Describe how the incremental model works
  1. Can it be plan driven, can it be iterative? You can iterate within increments, you can have increments planned
10. Describe Integration and Configuration process model
  1. What risks are reduced (development risk is reduced by reuse – instead lack of ownership of what is reused, introduces risk of not being able to make desired changes to reused parts, at all or in the desired time frame)



# Q4: Comparison of plandriven and agile SOE proces

## Example on what questions could be

1. What is the difference between plandriven and agile? (plandriven believe desired result can be predicted, agile expects change and uses frequent inspect and adapt to create best value)
2. How does Böhm/Turner define primary factors?
  1. Application (small, rapid change, turbulent environment)
  2. Management (onsite, qualitative control, tacit knowledge)
  3. Technical (Prioritized informal requirements, simple design,
  4. People (Cochkburn L2 and L3 developers)
3. Why do requirements change? (Business, technology, learning from use)
4. How can you manage requirements and changes to requirements
  1. Change process, analysis of impact
5. What is continuous integration and how does it relate to agile?
6. Prototype development
7. Agile
  1. XP practices
  2. Scrum roles
    1. PO+SM+Team
  3. Practices
    1. Sprint Planning+Daily Scrum+Sprint Review+Sprint Retrospective,+Baclog refinement
  4. Assets
    1. Product Burndown+Sprint Burndown+Scrum board
  5. Home grounds
8. Plandriven counterparts
  1. Lots of roles incl. Managers and specialists, knowledge sharing through documentation, Project Plan, Gant chart ...



# Q5: Key features of Scrum

## Example on what questions could be

1. What is Scrum
  1. Iterative agile method
2. Describe essential elements from Scrum
  1. Scrum roles, (Product Owner, Scrum Master, Team)
  2. Scrum practices (Sprint Planning+Daily Scrum+Sprint Review+Sprint Retrospective,+Backlog refinement)
  3. Scrum assets (fx Product Burndown+Sprint Burndown+Scrum board)
3. What is the focus of Scrum to the development process? (Focus on empirical instead of defined process, and therefore the three pillars of Scrum are Transparency, Inspect og Adapt)
4. Can you mention one or more Core values in Scrum:
  1. Commitment (to iteration goal), Focus (on iteration goals) , Openness (to work and progress) , Respect (or team responsibility) and Courage (for management to trust team, for team to take responsibility)
5. Can you mention some typical errors or mistakes in the use of Scrum
  1. Scrum master implemented as manager who tells team what to do (right way: Facilitator for team)
  2. Customers are not involved in each iteration
  3. New requirements or tasks are added to team during iteration
6. Can you say something about eXtreme Programming (XP) and what techniques that in particular fit to Scrum
  1. Fx Customer on-site, user stories, planning game, ...
7. What is the difference between plandriven and agile?
8. How defines Böhm/Turner the primary factors to balance plandriven and agile?
  1. Application (small, rapid change, turbulent environment)
  2. Management (onsite, qualitative control, tacit knowledge)
  3. Technical (Prioritized informal requirements, simple design,
  4. People (Cochburn L2 and L3 developers)
9. Why do requirements change? (Business, technology, læring fra brug)
10. How can you manage requirements and requirements change
  1. Change process, analysis of impact
11. What is continuous integration and how does it relate to agile
12. Prototype development, XP practices, Home grounds



# Q6: Key features of RUP – example questions

1. What is RUP
  1. Inkrementiel, iterative method framework
2. Describe essential elements of RUP
  1. Development in short timeboxede iterationer
  2. Develop high-risk and high value elements early, and preference for reuse of existing components
  3. Focus to deliver early value to customer
  4. Allow changes early in project
3. What phases are there in RUP (Note they are different from the disciplines/activities in waterfall: analys, design, development, validation, evolution)
  1. Inception, Elaboration, Construction, Transition
4. What disciplines are ingrained in a specific iteration
  1. Each phase has a different mix of the disciplines: requirements engineering, design, implementation, test and project management
5. What is tailored for the specific project from UP
  1. Practices, work products from a huge amount of selectable elements
6. Please mention examples on UP best practices (fx fra list of Six best practices)
  1. Timeboxed iterations
  2. Cohesive architecture striving to reuse existing components
  3. Continously verify quality
  4. Visual modelling
  5. Manage requirements
  6. Manage change
7. Does RUP have any explicit values?
  1. No, but you can derive a focus that's more projectoriented than for instance people oriented, e.g.
    1. Apply UP guidelines and best practices
    2. Be risk and value driven
    3. Establish clear vision
    4. It is critical to ensure that UP process is tailored to the project with the minimnal amount of work products
    5. A well defined process is useful
8. What are typical pitfall or misunderstandings
  1. RUP is not waterfall, the content of each phase (Inception, Elaboration, Construction, Transition) is a mix of the disciplines (analysis, design, implementation, test, ...). Iteration are longer than 6 weeks
9. Can you apply Scrum and XP with RUP? Yes
10. Can you say something about process improvement? Do you know other process frameworks? (Scrum, CMMI)



# Q7: Requirements Elicitation and Management

## Example questions

1. What main requirement activities are there (Elicitation and analysis **of needs**, specification of **requirements**, validation of **requirements**) (L1)
2. How can requirement activities be organized, and
  1. Spiral model is one way, and it uses activities to requirement validation, risk analysis, models and prototypes (L6 – last slide)
3. What are the steps in requirements elicitation
  1. A) Discovery & Classification B) Categorization C) Prioritization & Negotiation D) Documentation
4. Why is it difficult to elicit requirements?
  1. Many stakeholders with conflicting needs, stakeholders talk their own language, tacit knowledge and unconscious actions are not communicated, stakeholder and requirements engineer talk two different language
5. What techniques can be used to elicit requirements
  1. Interview (Open or closed)
  2. Etnography (Observe how work is performed)
  3. Prototypes (L1 – slide 31)
6. What is a recognized way to communicate requirements
  1. Stories / scenarios
7. How are requirements documented in Waterfall and in Scrum
  1. Waterfall: Approved requirements document with strict change management
  2. Scrum: Product vision and product backlog, reviewed and updated every sprint
8. How are requirements negotiated with stakeholders in waterfall and Scrum
  1. Waterfall: Up front in the requirements phase – state it now or it will be difficult later to get it
  2. Scrum: Ongoing refinement of product backlog with stakeholders, say what is most important now, we will continue

L=Lecture



# Q8: Managing Change – example questions

1. Why are changes managed in a project or product
  1. Primary: Manage how change in stakeholders needs drive changes to requirements
  2. Secondary: Manage configurations and versions of software releases
2. Who approves changes to requirements
  1. In waterfall this is often done by a project manager or a steering group for the project
  2. In agile the Product Owner has the mandate to approve changes, by updating the product backlog
3. What strategies can be used to reduce cost of rework due to requirements changes
  1. "Expect changes": and use fx prototyping
  2. "Tolerance to changes": design in a way where it is easy to deliver changes incrementally
4. What can prototypes be used for?
  1. Validation from users that a part of the solution is "fit for use".
5. What challenges are there to use prototypes?
  1. Users apply the prototype different from how they will work with the final system
  2. Those who test the prototype are not or do not correspond to the "real" end users
  3. Insufficient time to train users in the use of the prototype
6. What are advantages from incremental development
  1. Customers can use early increments as prototypes, but the increment is not thrown away, rather it is extended
  2. Customers can realise value from what is delivered early
  3. It is easy to incorporate changes while the system is being developed
7. What are disadvantages to incremental development
  1. Problematic when a complete system is replaced – impractical with an incomplete system together with the old system
  2. It can be difficult to define the shared base for the entire system, early in development
  3. Incremental/iterative approach develops and refines the requirements specification as we go – conflicts with contracts
8. How are changes to versions and configurations of code and releases managed? (Configuration Management)



# Q9: Quality Control - Verification and Validation

## Example questions

1. When we talk about quality, we often refer to two key concepts starting with V?
2. What is Verification and Validation?
  1. With respect to purpose?, Users expectations?, contracts?
3. What is inspection, and what is test and what is (peer) review?
4. What is test driven development (TDD)
  1. What is it good for? (code coverage, regressiontest, simplified debugging, documentation)
5. What is regressiontest?
6. What agile practices support V&V and what is a quality culture
  1. Definition of Done
  2. Sprint Review
  3. Check before check-in
  4. Never break the build
  5. Fix problems when you see them
  6. Culture = team members assumes responsibility for ensuring high quality
  7. XP: Customer on site
  8. XP: Pair programming



# Q10: Risk Management

## Example questions

1. What is a risk? Something that may happen and causes a loss
2. Example on risk categories
  1. Uncertainty, project, technical, business
  2. Keyperson from team dies, a supplier is not delivering as promised
3. How do you do risk analysis
  1. Identify risks and calculate risk exposure and describe consequence
    1. Risk exposure = probability \* loss, describe consequence
  2. Prioritize according to risk exposure, establish cut-line
  3. Establish for each risk above the cut-line (RMMM: risk mitigation, monitor, management)
    1. mitigation (prevention) plan – how can we prevent risk to happen
    2. contingency plan (management) plan – if it happens anyway, what do then do
    3. how to monitor development of the risk – called RMMM plan
4. How are risk management part of project management
  1. Waterfall / plan driven
    1. Risk and risk plans are part of the plans in project management
    2. Development of other plans contribute to identification of risk
    3. It is planned how
  2. Agile – inspect and adapt is reduction to produce the right product
    1. Daily Scrum: Do you have any impediments
    2. Sprint review: Inspects risk related to product and stakeholder
    3. Sprint retrospective: Adresses risks related to how the team works
5. What is the spiral model and how is it related to risk management?
6. What are Boehms primary risks (See spiral model article)
  1. ( Personel shortcomings, unrealistic schedule, wrong function, ..)



# Other topics: Boehm – Spiral Model – example questions

1. Hvad er spiralmodellen og hvordan indgår risikoanalyse?
2. Hvilke procesmodeller er spiralmodellen en reaktion på?
  1. Code-and-fix
  2. Stage-wise
  3. Vandfald
3. Hvad er Boehms fokus med spiral modellen
  1. At lave en model der er drevet af at styre risiko
4. Hvad er Böhms primære risici (spiralmodel article)
  1. (Personel shortcomings, unrealistic schedule, wrong function, ...)
5. Hvad har Boehms spiralmodel med riskostyring at gøre?
6. Kan du sætte modellen i perspektiv til requirement specification?
  1. Modellen beskriver konkrete aktiviteter der drives af en risikovurdering og kan gennemføres inkrementielt
7. Hvordan laver man en risikoanalyse
  1. Identify risk, exposure = probability \* loss, describe consequence
  2. Prioritize according to exposure, establish cut-line
  3. Establish mitigation (prevent) & contingency plan. Monitor
8. Hvordan indgår risk management i project management
  1. Risk er en af de planer der indgår i projektstyring
  2. Udarbejdelse af de øvrige planer bidrager med identifikation af risk



# Q11: Project Planning and Management

## Example questions

1. How is plandriven project planning
  1. Plan the work, work the plan, assume you can predict what is delivered
2. How is progress measured? Through milestones and related documentation
  1. Why are milestones important? Measure of progress according to plan
3. What is agile planning
  1. Welcomes changes, works from a prioritized product backlog, where content is constantly refined from the top and adjusted to learning
  2. What is sprint planning in Scrum, planning game in XP
4. Plandriven estimation
  1. Experience based estimation (likely, optimistic, pessimistic)
  2. Algoritmic based e.g. Cocomo – use of models and paramters
5. Agile estimation
  1. Based on velocity of team measured in story points



# Q12: Quality Management – example questions

Lx – Sn means see Lecture x slide n

1. What is quality (L7 – S10)
2. Explain some software qualities (L7 – S11)
3. What is a trade-off (L7 – S13)
4. What is total cost of quality and cost of correction? (L7 – S14-16)
5. What do we mean by subjective versus objective quality in Quality Management?
6. Hvad menes med subjektiv versus objektiv i Quality Management
7. Quality Management assumes good process → good product, or same process delivers same quality
8. What is a quality culture?
9. Hvad er quality review (peer review)?
10. How is quality ensured in Agile?
11. What is pair programming in XP and what are the results?
12. What is verification and validation? (L7 –S18)



# Q13: Configuration Management

1. Challenges in version management/control?
  1. Pessimistisk file locking, optimistic version merging
2. What is CM, branching, merging
  1. Techniques to support release, builds, baselines
3. What is the difference between centralize and distributed VC system and why is distributed useful
  1. Backup mechanism, off-line, project support default
4. What goes into a release?
  1. All code, data, configurations files, documentation
5. Agile: Continous integration
6. Daily build?
7. Plandriven and agile CM – advantages and disadvantages
  1. TDD and agile gives better code coverage, regression test suite are created incrementally, simplifies debugging, test also documents code



# Other topics: Software Process Improvement

- Ian Sommerville Kap 2.4 – To tilgange til procesforbedring
  - Proces modenhed fx med brug af CMMI
  - Agil – er i hele sin tilgang fokuseret på mindst mulig procesoverhead
- En organisations tilgang til procesforbedring kan være
  - Plandrevet – CMMI (Systematic case)
    - Gør det der står i modellen
  - Inkrementielt – målingsbaseret (B&K case)
    - Baseret på måling og eksterne måledata, lav en forbedring af gangen og mål effekten efter hver
- Kultur – hvilken kultur er der i virksomheden og hvordan påvirker kulturen fokus på procesforbedring (Umoden kultur hylder brandslukkere, moden kultur hylder dem der følger processen.)
- CMMI – Første skridt fra 1 til 2 handler om at indføre projektledelse discipliner
- Agile: I scrum bruger man hvert sprint retrospective til at overveje forbedringer af processen

