

1.43 Lean Software Development

- **Course name:** Lean Software Development
- **Course number:** XYZ

1.43.1 Course characteristics

1.43.1.1 Key concepts of the class

- Fundamental principles of producing software as a creative act of the human mind
- Techniques to optimize such production, with specific focus on agile methods

1.43.1.2 What is the purpose of this course?

This course exposes the student to the core concepts behind Lean Development in Software Engineering, beyond myths and legends, emphasizing how it relates to the general principles of Lean Development. It discusses the different possible software processes, how they can be tailored, enacted, and measured. In addition, a significant part of the course is centered around the application of lean to software development to knowledge intensive areas not necessarily connected to software.

1.43.1.3 Course Objectives Based on Bloom's Taxonomy

- What should a student remember at the end of the course?

By the end of the course, the students should be able to recognize and define:

- creative nature of software production as an act of creativity of the human mind,
- the substantial differences between tame and wicked problems,
- the core concepts of measurement in software engineering,
- the fundamentals of Taylorist/Fordist approaches to (software) production,
- the basis of lean and agile software development,
- the “dark” side of agility,
- the importance of knowledge and knowledge sharing in producing software,
- how to create an ad-hoc process for a development organization.

- What should a student be able to understand at the end of the course?

By the end of the course, the students should be able to describe and explain (with examples):

- when a problem is “easy to solve” provided enough effort is put to such solution and when it is not,
- what is a measure in general,
- why it is important and how we can define and perform measurements in software engineering, especially in lean and agile development environments,
- how to organize the development process to collect metrics non invasively,
- the difference between pulling and pushing in (software) development,
- the fundamental principle of agility,
- the risk intrinsic in the dark side of agile,
- how to organize an agile development process based on the definition of overall Goals, associated Question, and milestones based on Metrics,
- an environment based on experience, like the Experience Factory.

- What should a student be able to apply at the end of the course?

By the end of the course, the students should be able to:

- compute the fundamental software metrics to track the evolution of a project,
- organize the aims of a (software) development organization in terms of Goals, Questions, and Metrics,
- create a tailored (lean and agile) development process for an organization producing software,
- define a path to insert and manage such (lean and agile) development process into an organization producing software,
- structure the experience gathering during inside an organization to based on it the future strategic decision of such organization,
- relate the various proposals for Agile Methods to the overall principles of Lean Management,
- define a suitable (lean) process for a new organization, a process to introduce and institutionalize it, and an approach to measure the outcome of such introduction and institutionalization.

1.43.1.4 Course evaluation

Each component apart from the weekly tests will be assessed on a scale 0-10, where 6 is the minimum passing grade. In case of exceptional work a 10 cum laude will be assigned, with a numeric value from 11 to 13 at discretion of the instructor. The homeworks will be initially

Table 1.110: Course grade breakdown

Type	Default points	Proposed points
Labs/seminar classes (weekly evaluations)	20	35 ^a
Interim performance assessment	30	(merged with the above)
Exams	50	65 ^b

^aPlease note that Labs/seminar classes (weekly evaluations) is divided in 10% for class participation and weekly tests, 10% for the Weekly GQM and 15% for invited speaker presentations, questions, and answers.

^bOf which 25% for the report on the application of Lean Principles to artistic areas, 10% for the overall GQM, and 30% for the oral.

graded on a scale 0-2 weekly and then the overall grade will be assembled on a scale 0-10. The grading, though, is not a simple linear combination of the components above. In particular:

- failing any part of the evaluation will trigger a failure in the entire course,
- the weekly quizzes will be averaged on the (n-4) best performances, where the 4 will be used to consider any kind of absence of the student, without requiring any additional documentation; if the student has legitimate reasons to be absent more than 4 times, then supporting document should be provided for all her/his absences,
- if there are not failing components, the final grade will be computed as a weighted average of the components above approximated at the highest second digit and then rounded to the closest integer.

Note that the questions for the exam are taken from the textbook.

1.43.1.5 Grades range

Table 1.111: Course grading range

Grade	Default range	Proposed range
A. Excellent	90-100	96-100
B. Good	75-89	66-95
C. Satisfactory	60-74	56-65
D. Poor	0-59	0-55

The semester starts with the default range as proposed in the Table 1.111, but it may change slightly (usually reduced) depending on how the semester progresses. See above.

1.43.1.6 Resources and reference material

- **Textbook:** Andrea Janes and Giancarlo Succi. Lean Software Development in Action. Springer, Heidelberg, Germany, 2014. ISBN 978-3-662-44178-7. doi: 10.1007/978-3-642-00503-9. URL <http://dx.doi.org/10.1007/978-3-642-00503-9>
- **Reference:** James P. Womack and Daniel T. Jones. Lean Thinking: banish waste and create wealth in your corporation. Lean Enterprise Institute. Simon & Schuster, 1996. ISBN 9780684810355. URL <https://books.google.ru/books?id=DJwoAQAAMAAJ>
- **Reference:** Taiichi Ohno. Toyota production system: beyond large-scale production. CRC Press, 1988
- **Reference:** James P. Womack. Lean Thinking. Simon & Schuster, Limited, 1997. ISBN 9780671004712. URL <https://books.google.ru/books?id=4mJMPwAACAAJ>
- **Reference:** James P. Womack, Daniel T. Jones, and Daniel Roos. The Machine That Changed the World: The Story of Lean Production. Harper Perennial modern classics. HarperCollins, 1991. ISBN 9780060974176. URL <https://books.google.ru/books?id=Jz4zog27W7gC>
- **Reference:** M. Poppendieck and T. Poppendieck. Implementing Lean Software Development: From Concept to Cash. A Kent Beck signature book. Addison-Wesley, 2007. ISBN 9780321437389. URL <https://books.google.ru/books?id=3TM0AgAAQBAJ>
- **Reference:** C.M. Christensen. The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail. Management of innovation and change series. Harvard Business School Press, 1997. ISBN 9780875845852. URL https://books.google.ru/books?id=SIexi_qgq2gC

1.43.2 Course Sections

The course is organized in 15 weeks with every weeks 2 academics hours of lectures and 4 academic hours of labs. The main sections of the course and approximate hour distribution between them is as follows:

Table 1.112: Course Sections

Section	Section Title	Teaching Hours
1	Software as a creative activity	12
2	Measurement in software	24
3	Taylorism and Fordism	6
4	Lean and Agile	24
5	Issues in Lean and Agile	6
6	Structuring a Lean Approach to software development	6
7	Optimizing the development process	12

1.43.2.1 Section 1

Section title:

Software as a creative activity

Topics covered in this section:

- Nature of software
- Software and art
- Core resources for the production of software
- Tame and wicked projects
- Organizing the activities based on the GQM approach

What forms of evaluation were used to test students' performance in this section?

Form	Yes/No
Development of individual parts of software product code	0
Homework and group projects	1
Midterm evaluation	0
Testing (written or computer based)	0
Reports	1
Essays	0
Oral polls	1
Discussions	1

Typical questions for ongoing performance evaluation within this section

1. Provide examples of creativity in the production of software.
2. Describe the differences between tame and wicked projects.
3. Discuss the key resources needed for the production of software.
4. What are key issues in creative production of software for distributed teams?

Typical questions for seminar classes (labs) within this section

1. Provide examples of wicked problems from your everyday life
2. Evidence wickedness in different aspects of software production
3. Create a GQM for your aims of the semester
4. Discuss the role of GQM in tame and wicked projects

Test questions for final assessment in this section

1. Present the salient aspects of wicked problems
2. List the key aspects of software that make it a wicked problem
3. Provide a link between wickedness and creativity in software production

4. Discuss what promotes and what inhibit creativity in general and in software production
5. Outline meaning and limitation of the concept of “engineering” the production of software

1.43.2.2 Section 2

Section title:

Measurement in software

Topics covered in this section:

- Meaning of measures
- The representational theory of measurement
- Measurement scales
- Fundamental measures for the production of software
- Procedural measures
- Object Oriented measures

What forms of evaluation were used to test students' performance in this section?

Form	Yes/No
Development of individual parts of software product code	0
Homework and group projects	1
Midterm evaluation	0
Testing (written or computer based)	0
Reports	1
Essays	0
Oral polls	1
Discussions	1

Typical questions for ongoing performance evaluation within this section

1. Provide examples of the representational theory of measurement
2. List the measurement scales
3. Present for each measurement scale the operations that can be performed on it
4. Discuss the representational condition
5. What are key size metrics?
6. What are key complexity metrics?

Typical questions for seminar classes (labs) within this section

1. Provide examples of subjective and objective metrics

2. List 3 direct and 3 indirect measures, evidencing also the problems connected to the construction of indirect measures
3. Compute LOC and other size metrics for code snippets
4. Compute MCC and other complexity metrics for code snippets
5. Compute the metrics of the CK suite for portions of Object Oriented systems

Test questions for final assessment in this section

1. Structure the aims of a company using the GQM and detailing the metrics to compute, also explaining the deductions and the predictions that can be made with such metrics.
2. Given an index computed as a combination of metrics, determine if it is a metrics according to the representational theory of measurement.
3. Analyse a portion of a system and determine suitable metrics to extract and the information that would be provided by such metrics.
4. Given a website performing a service (like flight reservations), compute the Function Points for such website.
5. Discuss how to structure a taxonomy of quality for a specific company, explaining the role of reliability in it, and detailing how to compute the reliability again in the context of such company; please make the assumptions that you need to perform such computation.

1.43.2.3 Section 3**Section title:**

Taylorism and Fordism

Topics covered in this section:

- The increase of productivity in the idea of Taylor
- The role of division of work
- Planning and formalization of tasks
- Economies of scale
- Problems in understanding tasks
- Taylorism/Fordism and software development

What forms of evaluation were used to test students' performance in this section?

Form	Yes/No
Development of individual parts of software product code	0
Homework and group projects	1
Midterm evaluation	0
Testing (written or computer based)	0
Reports	1
Essays	0
Oral polls	1
Discussions	1

Typical questions for ongoing performance evaluation within this section

1. Detail the fundamental assumptions by Taylor
2. What are the fundamental activities of managers according to Taylor?
3. In which sense Taylor has influenced what we now consider "good management practices?"
4. How does creativity relates to the "good management practices" of Taylor?
5. What are the problems in applying Fordism/Taylorism to software development?

Typical questions for seminar classes (labs) within this section

1. Provide examples of companies where the approach by Taylor has been successful
2. Discuss how the approach of Taylor can be useful in attracting and/or retaining employees
3. Analyse the fundamental activities of managers according to Taylor and determine their limits
4. Determine the fundamental activities of managers according to Taylor can have an implication in the software crisis
5. Outline how flexibility and variable requirements can be handled in the context of Taylorism and Fordism

Test questions for final assessment in this section

1. What aspects of making software feel like an art, which like a craft to you and how could you bend Taylorism and Fordism to handle it?
2. How does Fordism and Taylorism can explain that companies producing software containing bugs can still stay in the market, and, in certain case, also be successful?
3. Based on your experience and previous courses, which development models can refer to Fordism and Taylorism and which cannot be reduced to it?
4. How does the specialization of work in software development can be linked to Fordism and Taylorism?

5. Provide an example for each of the fundamental activities of managers according to Taylor that shows how such activity is very useful in software development and one that shows that is inadequate.

1.43.2.4 Section 4

Section title:

Lean and Agile

Topics covered in this section:

- Taiichi Ono and the Toyota Production System
- Creating a “Radiography” of the Production Process
- Workers involvement
- “Pull” and Not “Push”
- Kanban
- Quality management
- Process control
- Job enrichment
- Control and coordination mechanisms
- Case study: Extreme Programming

What forms of evaluation were used to test students’ performance in this section?

Form	Yes/No
Development of individual parts of software product code	0
Homework and group projects	1
Midterm evaluation	0
Testing (written or computer based)	0
Reports	1
Essays	0
Oral polls	1
Discussions	1

Typical questions for ongoing performance evaluation within this section

1. Present the key problems in batch production
2. Outline the key principles of the approach of Ono at Toyota
3. What are the fundamental steps in eliminating waste according to Ono?
4. Details the role of the customers and of the workers in the approach of Ono
5. Explain the difference between “Pulling” and “Pushing”
6. What are key steps in improving quality according to Ono?
7. What are the key control and coordination mechanisms available?

Typical questions for seminar classes (labs) within this section

1. What are the fundamental two actions needed to perform a Lean transformation according to Ono?
2. What are the associated three major needs?
3. What are the 5 steps to enact Lean Thinking according to Womak and Jones?
4. Discuss the 8 constantly ongoing activities in a lean company like Toyota
5. How can activities been classified in a decision matrix in an environment like Toyota?
6. Provide concrete examples of “Push” and of “Pull” in software production.
7. Details the control and coordination mechanisms present in agile and in traditional development environments.

Test questions for final assessment in this section

1. Structure a model like PDSA for a company producing websites for online marketing.
2. How can the right part and the right information be always available without waste according to the Toyota approach?
3. Compare a street crossing based on traffic lights with a roundabout and determine the approach that is safest and the one with the a highest throughput according to Waterfall and to Lean.
4. Discuss the involvements of workers in tayloristic/fordistic and in Lean development processes and their implications for the retention and the improvement of the quality of the workforce.
5. Outline the extent of which economies of scale exist in Lean development processes.
6. Imagine you had to introduce Extreme Programming in a software development team that follows a waterfall process. Which problems do you foresee? How will the clients react (that until now are used to work with a team that used the waterfall process)? How would you address them?

1.43.2.5 Section 5**Section title:**

Issues in Lean and Agile

Topics covered in this section:

- The “Hype of Agile”
- The dark side of agile

- Skepticism about agile methods
- Knowledge and software engineering
- Using burn-down charts
- The Zen of agile

What forms of evaluation were used to test students' performance in this section?

Form	Yes/No
Development of individual parts of software product code	0
Homework and group projects	1
Midterm evaluation	0
Testing (written or computer based)	0
Reports	1
Essays	0
Oral polls	1
Discussions	1

Typical questions for ongoing performance evaluation within this section

1. What is the Gartner's Innovation Hype Cycle? Could you provide examples of application of it to a different field of knowledge?
2. Describe the so-called "Dark Agile Manifesto"
3. What are the sources of the skepticism present with respect to Agile?
4. What makes agile awkward in the eyes of "traditional" managers?
5. Given a production system, how do you determine what is the value of every step, how much improvement can be considered enough, and when is the point reached where value is not increased anymore but destroyed?
6. How is it possible to obtain knowledge about the production process? How can I create visibility of the ongoing activities or problems?

Typical questions for seminar classes (labs) within this section

1. Identify in other areas of software engineering phenomenon similar to the "Dark Side of Agile."
2. Identify in other knowledge-intensive fields phenomena similar to the "Dark Side of Agile" and discuss how they can be tackled.
3. Given a production process, determine strategies to store knowledge to create experience?
4. How can you design the production process so that the team uses the gained experience?
5. How can you systematically improve your process, also building on the experience to anticipate problems (create wisdom)?

Test questions for final assessment in this section

1. Imagine you are the boss of a small software development company. Which actions would you do or which practices would you introduce to prevent that your programmers fall into the trap of following a software guru?
2. Elaborate possible extreme (and damaging) positions that can be taken by gurus of agile.
3. Explain the differences in introducing a methodology by a guru and by a smart and effective coach.
4. Detail why Extreme Programming produces an informative workspace.
5. Discuss effective ways of packaging and “distributing” knowledge in software teams, starting with the guru approach of organizing knowledge into simple, clear practices which are easy to explain and to follow.

1.43.2.6 Section 6

Section title:

Structuring a Lean Approach to software development

Topics covered in this section:

- Existing proposals to create a “Lean Software Development”
- Sharing a common vision
- Depriving gurus of their power
- GQM+
- Applying the GQM+ step-by-step
- Business alignment
- GQM+ for business alignment

What forms of evaluation were used to test students’ performance in this section?

Form	Yes/No
Development of individual parts of software product code	0
Homework and group projects	1
Midterm evaluation	0
Testing (written or computer based)	0
Reports	1
Essays	0
Oral polls	1
Discussions	1

Typical questions for ongoing performance evaluation within this section

1. Reflect on the seven principles for characterizing Lean Software Development by Mary and Tom Poppendieck.
2. What is the semantic gap and which threats it poses to effective software development?
3. Map the structure of Extreme Programming to the layered structure of Shalloway et al.
4. How does the scientific method deprives gurus from their power?
5. What are the steps to implement the GQM+?
6. For what reason measurement goals and business goals should be interconnected?

Typical questions for seminar classes (labs) within this section

1. How do the practices of Lean Management defined by Hibbs and colleagues relate to the seven principles by Mary and Tom Poppendieck
2. Propose how you could develop software and hardware tools to promote common visions in companies.
3. Elaborate a proposal to create some kind of Balanced Scorecards to evaluate your current study.
4. Discuss the opinion of Ono about following plans and the extent to which such opinion contradicts (a) the practices, and (b) the principles of the tayloristic/fordistic approach
5. Propose a SWAT analysis to introduce a Lean approach to your most recent software development endeavour

Test questions for final assessment in this section

1. Discuss the role of customer on-site under the perspective of lean and outline its relevance in the earlier and then in the later proposals of agile software development.
2. Suppose that you have to develop Balanced Scorecards for a software development team. Which perspectives would you use? Which goals would you use for each perspective?
3. What is a socio-technical system and how can it be used to describe an (agile) software production environment?
4. Imagine you want to evaluate how readable the source code of some program is. Define a GQM+ model to describe what and why you would measure.
5. The development in company M occurs according to the following schema: when a new project is started, a developer takes an old project that is the most similar to the new requirements and makes a copy and starts implementing the required

modifications. To improve this process and to help the company to adopt a component-based approach, we want to understand which pieces of code are the best candidates for future components and which variability points they have. Define a suitable GQM+ for such purpose.

1.43.2.7 Section 7

Section title:

Optimizing the development process

Topics covered in this section:

- Why the PDSA does not work in software
- The experience factory
- The QIP cycle
- Non invasive measurement
- The big-brother effect
- The role of automation
- Employing Andon boards

What forms of evaluation were used to test students' performance in this section?

Form	Yes/No
Development of individual parts of software product code	0
Homework and group projects	1
Midterm evaluation	0
Testing (written or computer based)	0
Reports	1
Essays	0
Oral polls	1
Discussions	1

Typical questions for ongoing performance evaluation within this section

1. Details the major components of an experience factory.
2. What are Reflection, Retrospective, and Post-Mortem Analysis? Why are they useful in Lean Software Development?
3. What are the key components of a non invasive software measurement systems?
4. What is the big brother effect and how it is possible to alleviate it.
5. Discuss how automation is present in Extreme Programming.
6. Where is the term "Dashboard" coming from and what is its use in Lean Software Development?

Typical questions for seminar classes (labs) within this section

1. List the steps of a QIP.
2. Discuss the risks of a measurement program and how non invasive software measurement can help alleviating them.
3. Are there cases in which Theory X of management could be more effective than Theory Y? Discuss your findings.
4. For which aspects of software production automation could be useful?
5. Which tools could be used to promote automation?
6. Prototype by paper and pencil an Andon board that you would consider useful in a software production environment.

Test questions for final assessment in this section

1. What type of wisdom (in the sense of “know-why”) would you manage in an Experience Factory to support Lean Thinking? Distinguish between organizational learning and project learning.
2. Assume you are a manager convinced that Theory X is true. Which non-invasive measurement probes would you want to develop to maximize productivity? Now assume you are convinced that Theory Y is true. Which non-invasive measurement probes would you need now?
3. We discussed that we foresee two ways to collect measurements non- invasively: in batch and in background mode. What are the advantages and disadvantages of each approach?
4. There are many interrelated building blocks (or concepts) of Lean Software Development and each contributes differently to it. What types of data are handled by each of these building blocks (or concepts)? How do the contribution to the overall value stream, to the creation of knowledge, and to the overall improvement?
5. Assume you set up a fantastic dashboard for your team. As you collect the data and visualize it, you notice that all the measurements show problematic values. You let the dashboard in place for some days and also show it to your collaborators, but nobody cares; everybody continues his job as if everything would be fine. What is going wrong?