DevOps – LTAT.06.015 – Lecture 1

Introduction to DevOps - part 1

06th Sept 2022

Chinmaya Dehury

chinmaya.dehury@ut.ee
OUTLINE

Part 1
• Software Development Life Cycle
• Waterfall model
• Agile model
• DevOps Motivation

Part 2
• DevOps history & stats
• ....many more on DevOps
Know audience

Quick Survey

https://www.menti.com
Code: 7025 3871

1. DevOps Experience
2. Which tools/concepts are you familiar with?
Software Development Life Cycle (SDLC)

• A lifecycle covers all the stages of software from its inception with requirements definition through to fielding and maintenance [3].

• E.g. Banking application
  - Front-end
  - Back-end
  - Databases
  - Analytics
  - endpoints

E.g. DevOps course
  - several lectures
  - practice sessions
  - grading
  - communication
  - resources
Software Development Life Cycle (SDLC)

• A lifecycle covers all the stages of software from its inception with requirements definition through to fielding and maintenance [3].
Software Development Life Cycle (SDLC)

• A lifecycle covers all the stages of software from its inception with requirements definition through to fielding and maintenance [3].
Software Development Life Cycle (SDLC)

• A lifecycle covers all the stages of software from its inception with requirements definition through to fielding and maintenance [3].

Anything missing?
Software Development Life Cycle (SDLC)

• A lifecycle covers all the stages of software from its inception with requirements definition through to fielding and maintenance [3].

Can you arrange them?

Icon src: https://www.iconfinder.com
Software Development Life Cycle (SDLC)

SDLC models

- Waterfall Model
- Spiral Model
- Agile Model
- Incremental model

Software Development Life Cycle (SDLC)

Waterfall Models

• Defined as early as ????

Implementation steps to deliver a small computer program for internal operations
Software Development Life Cycle (SDLC)

Waterfall Models

• Defined as early as 1970 by Dr. Winston W. Rovce [royce1970, Davis1988]

• Later improved in 1976 by Barry W. Boehm [Davis1988, boehm1976]

Implementation steps to deliver a small computer program for internal operations [royce1970]
Software Development Life Cycle (SDLC)

Waterfall Models

• Defined as early as 1970 by Dr. Winston W. Royce [royce1970, Davis1988]

• Later improved in 1976 by Barry W. Boehm [Davis1988, boehm1976]

Implementation steps to deliver a small computer program for internal operations [royce1970]
Software Development Life Cycle (SDLC)

Waterfall Models

- **Most basic stages** [Davis et al.]
  - System requirements
  - Software requirements
  - Preliminary Design
  - Detailed Design
  - Code and Debug
  - Test and pre-operation
  - Operations and maintenance

Software Development Life Cycle (SDLC)

Waterfall Models – General stages

Client

Gather the requirements and defining them (users, login, dashboard, purchase cart, payment options)
Language, DB, dev platform, other technical requirements...

Programmer

Writing the code, detecting and removing of existing and potential errors
report, monitor, resolve and retest software components until they reach the quality standards...

Planning/Designing

deals with algorithm design, software architecture design, database conceptual schema, logical diagram design, data structure definition, etc.

Coding & Debugging

Deliver and deploy the software, ready for production. (e.g. deliver the car...😊)

Testing

Make sure that the customized car is running smoothly....😊

Operation

Maintenance

Software Development Life Cycle (SDLC)

Waterfall Models - **Advantages**

- Simple and easy understanding and implementation
- Very structured organization
- Define -> Design -> Code
- Phases are very clear
- Easy to manage, arrange tasks
- Focus on one at a time
- Easy and well documentation
- Very straightforward steps

E.g. Delivery of DevOps course lectures

Software Development Life Cycle (SDLC)

Waterfall Models

1. **Client**
   - New Requirements
   - Programmer
   - Planning/Designing
   - Coding & Debugging
   - Testing
   - Operation
   - Maintenance

2. **Programmer**
   - Planning/Designing
   - Coding & Debugging
   - Testing
   - Operation
   - Maintenance

3. **Planning/Designing**
   - Coding & Debugging
   - Testing
   - Operation
   - Maintenance

4. **Coding & Debugging**
   - Testing
   - Operation
   - Maintenance

5. **Testing**
   - Operation
   - Maintenance

6. **Operation**
   - Maintenance

7. **Maintenance**

---

*No client is involved*
Software Development Life Cycle (SDLC)

Waterfall Model - **Disadvantages**

- Known requirements (Client is clear regarding their requirements)
- Slow delivery speed
- Very expensive to reflect the new requirements
- No concurrency (just think about a computer with single-thread computation)
- Risky and uncertainty, as no client is involved in other phases
- No room for requirement modification in later stage
Software Development Life Cycle (SDLC)

Agile Model

Requirements

Planning/Designing

Coding & Debugging

Testing

Operation

Maintenance

Client

Programmer

Ask for feedback

Client's feedback

Client
Software Development Life Cycle (SDLC)

Agile Model

**agile**

*adjective*

able to move about quickly and easily:

able to move your body quickly and easily:

able to deal with new situations or changes quickly and successfully:

**agile (MENTALLY)**

able to think quickly and clearly:

**agile (PHYSICALLY)**

used for describing ways of planning and doing work in which it is understood that making changes as they are needed is an important part of the job

Src: [https://dictionary.cambridge.org/dictionary/english/agile](https://dictionary.cambridge.org/dictionary/english/agile)
Software Development Life Cycle (SDLC)

Agile Model

• Address the challenge of an unpredictable world
• Focus on individuals and their creativity
  • rather than on processes and tools
• Promote quick response to
  • changing environments
  • changes in user requirements

Its all about “about feedback and change” [Williams2003]
Software Development Life Cycle (SDLC)

Agile Model
Software Development Life Cycle (SDLC)

Agile Model

Agile : Iteration 1

Planning ➔ Designing ➔ Develop ➔ Testing ➔ Feedback

Aprox. 2 weeks

Agile : Iteration 2

Planning ➔ Designing ➔ Develop ➔ Testing ➔ Feedback

Aprox. 2 weeks

Agile : Iteration 3

Planning ➔ Designing ➔ Develop ➔ Testing ➔ Feedback

Aprox. 2 weeks

...
What makes a development method an agile one? [6]

- **Breakdown** the whole application to different sub-services
- **Incremental**: small software releases, with rapid cycles
- **cooperative**: customer and developers working constantly together with close communication
- **Straightforward**: the method itself is easy to learn and to modify, well documented
- **adaptive**: able to make last moment changes
- **No monolithic application development**

Software Development Life Cycle (SDLC)

Agile Model – Principles

**Principles** of Agile development model [5]

1. Client’s satisfaction becomes higher priority
2. Welcome changing requirements
3. Frequent interaction with clients
4. Frequent working software delivery
5. Provide motivated environment and support to individuals
6. Face-to-face conversation (now probably on virtual mode)
7. Working software becomes the primary measure of progress
8. Sustainable development
9. Continuous attention to technical excellence and good design enhances agility
10. Simplicity – Maximize the amount of work not done
11. Frequently improve the effectiveness rather just focusing on fine-tuning and adjusting
12. The best architectures, requirements, and designs emerge from self-organizing teams

Src: [http://agilemanifesto.org/principles.html](http://agilemanifesto.org/principles.html)
Software Development Life Cycle (SDLC)

Agile Model - Advantages

1. Persistent software delivery
2. Quicker release through fixed, reliable and short-term schedules
3. Quick adoption of new features
4. Increased client’s satisfaction
5. Frequent interaction
6. Continuous improvement
7. Increased transparency through tighter collaboration
8. Improve product quality and predictability
9. Reduced risks
Software Development Life Cycle (SDLC)

Agile Model – existing methods

Agile Methods [6]

1. Extreme Programming
2. Scrum
3. Crystal family of methodologies
4. Feature Driven Development
5. The Rational Unified Process
6. Dynamic Systems Development Method
7. Adaptive Software Development
Software Development Life Cycle (SDLC)

Seems this is a perfect model for software development. then why we need DevOps?
Software Development Life Cycle (SDLC)

Agile Model:

Disadvantages:

1. Developer computer is used mainly for testing of those features

2. Test and production environment configuration mismatch

3. Usually not tested in production environment

4. Developers team and operations team are in silos

5. Continuous involvement of all the stakeholders.
Software Development Life Cycle (SDLC)

Agile Model:

Disadvantages:

Developers team → Developed product → Operations team → Product deployment → Production environment

Send product back to developer → Failed Product
Software Development Life Cycle (SDLC)

Agile Model:

Disadvantages:

- Agile addresses gaps in Customer and Developer communications
- DevOps addresses gaps in Developer and IT Operations communications

DevOps
Development + Operations

A culture
Development + Operations

Foster the culture
Focus on Outcomes
DevOps History

• Concept of DevOps emerged out of a discussion between Andrew Clay and Patrick Debois in 2008. Read more at: https://www.appknox.com/blog/history-of-devops

• Devopsdays event:
  • a worldwide series of technical conferences
  • Topics: software development, IT infrastructure operations, and the intersection between them.
  • Started 2009
  • A series of event in this month (https://devopsdays.org/events)
* this is not an advertisement
End of Part-1
Lab Sessions

- About the ETAIS marketplace
- Get familiar with creating, deleting and managing VMs
- Invitation to ETAIS DevOps course organization
- Communication over Zulip
  - Login: https://zulip.cs.ut.ee
  - Make sure that you are in DevOps2022Fall Zulip stream.
References


Any Question?

THANK YOU