Advanced Software Engineering Course

Software Development Process

Sadegh Sulaimany

info@Bioinfotmation.ir

دانشگاه کردستان
University of Kurdistan
زانکۆی کوردستان

SBNA
Social & Biological Network Analysis Laboratory

Initial assessment

1. What are the differences between Waterfall and RUP process models?

Agenda

- Review the Plan & Document software process models
  - Waterfall
  - RAD
  - Spiral
  - RUP

Successful software engineering

VS

amazon

- Obama Affordable Care Act (ACA)
  - Disaster!

Topic	Amazon.com	ACA Oct	ACA Nov	ACA Dec
Customers/Day (Goal)	–	50,000	50,000	30,000
Customers/Day (Actual)	>10,000,000	800	3,700	34,300
Average Response time (seconds)	0.2	8	1	1
Downtime/Month (hours)	0.07	446	107	36
Availability (% up)	99.99%	40%	85%	95%
Error Rate	–	10%	10%	–
Secure	Yes	No	No	No

Software Development Process

- Plan & Document
  - Start from 1960
  - planning and documenting approach
    - had worked well in other “big” engineering projects
    - such as civil engineering
  - Bring an engineering discipline for software engineering
    - Maximizing Q&P
      - Before starting to code, come up with a plan for the project,
        - including extensive, detailed documentation of all phases of that plan
      - Progress is then measured against the plan.
      - Changes to the project must be reflected in the documentation and possibly to the plan

Old mindset

Documentation should be written at all stages of development, and includes requirements, designs, user manuals, instructions for testers and project plans.

—Timothy Lethbridge and Robert Laganiere, 2002

Documentation is the lifeblood of software engineering.

—Eric Braude, 2001

- IEEE/ANSI standard 830/1993
  - SRS

Early version of Plan & Document

- Waterfall model
  - 1970

1. Requirements analysis and specification
2. Architectural design
3. Implementation and Integration
4. Verification
5. Operation and Maintenance

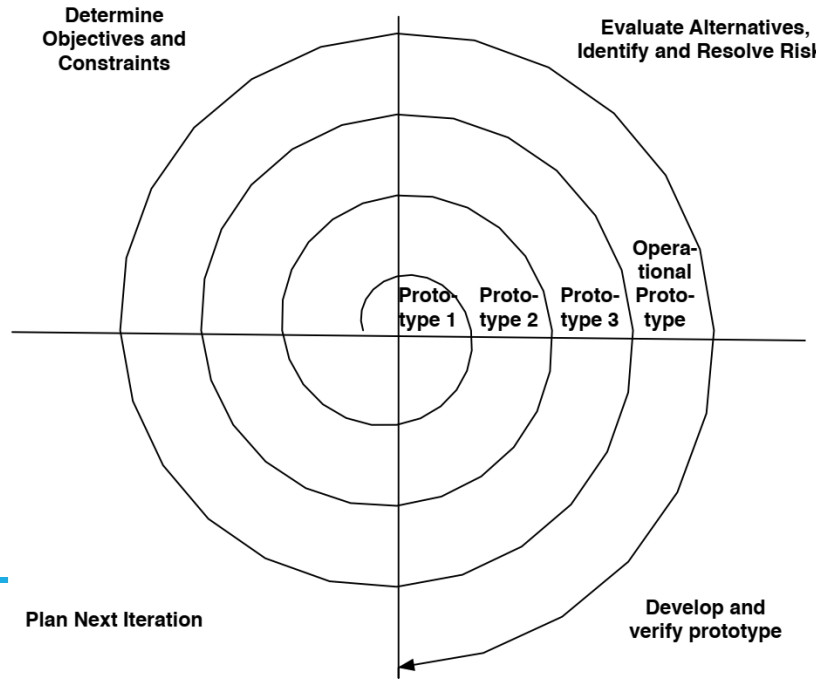
- Last typically between 6 to 18 months
- Idea: Earlier you find an error the cheaper it is to fix
  - Snowball effect
- important information is not lost if a person leaves the project

Early version of Plan & Document

- Waterfall model
  - When does it work well?
    - can work well with well-specified tasks
      - like NASA space flights
    - But
      - runs into trouble when customers change their minds

Other Plan & Document process models

- Spiral model
  - 1980
  - Idea:
    - it's easier for customers to understand what they want once they see a prototype
    - and for engineers to understand how to build it better once they've done it the first time



Other Plan & Document process models

- Spiral model
  - reduces chances of misunderstandings
  - iterations were 6 to 24 months long
    - still relies on planning and extensive documentation, but the plan is expected to evolve on each iteration

Other Plan & Document process models

- RUP
  - 1990
    - more closely allied to business issues than to technical issues
      - Like Waterfall and Spiral, RUP has phases
    - Object Oriented paradigm

1. Inception: makes the business case for the software and scopes the project to set the schedule and budget, which is used to judge progress and justify expenditures, and initial assessment of risks to schedule and budget.

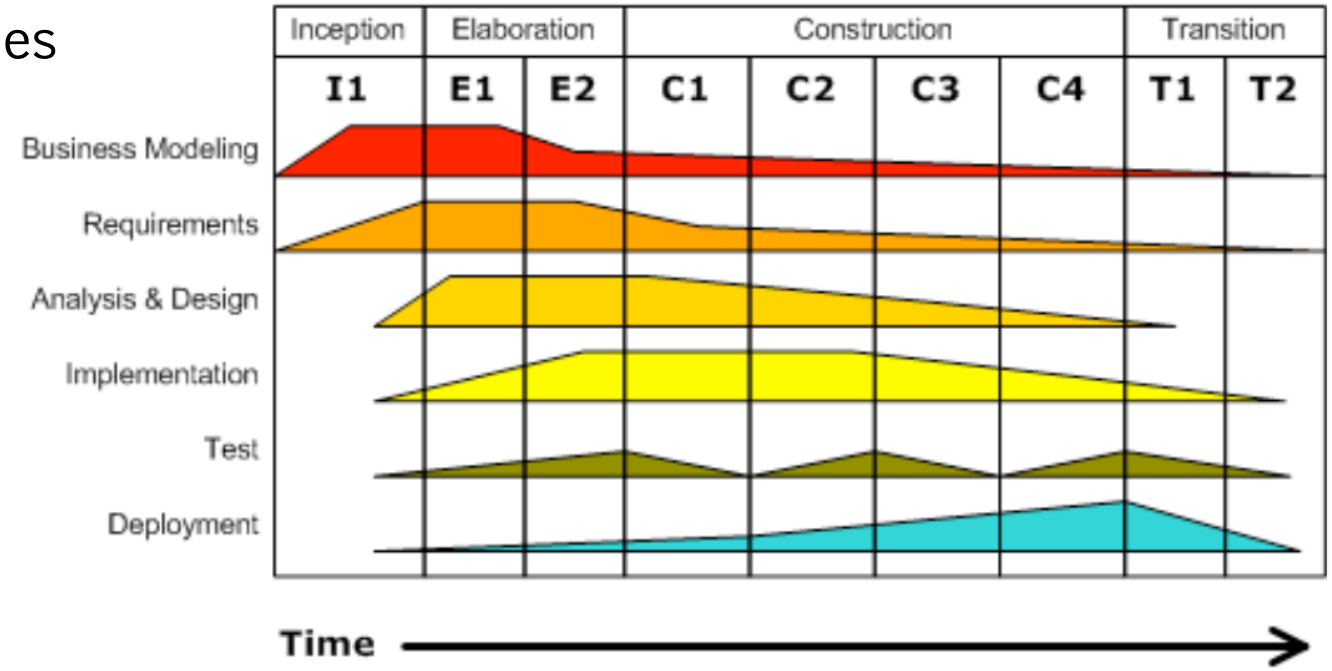
2. Elaboration: works with stakeholders to identify use cases, designs a software architecture, sets the development plan, and builds an initial prototype.

3. Construction: codes and tests the product, resulting in the first external release.

4. Transition: moves the product from development to production in the real environment, including customer acceptance testing and user training.

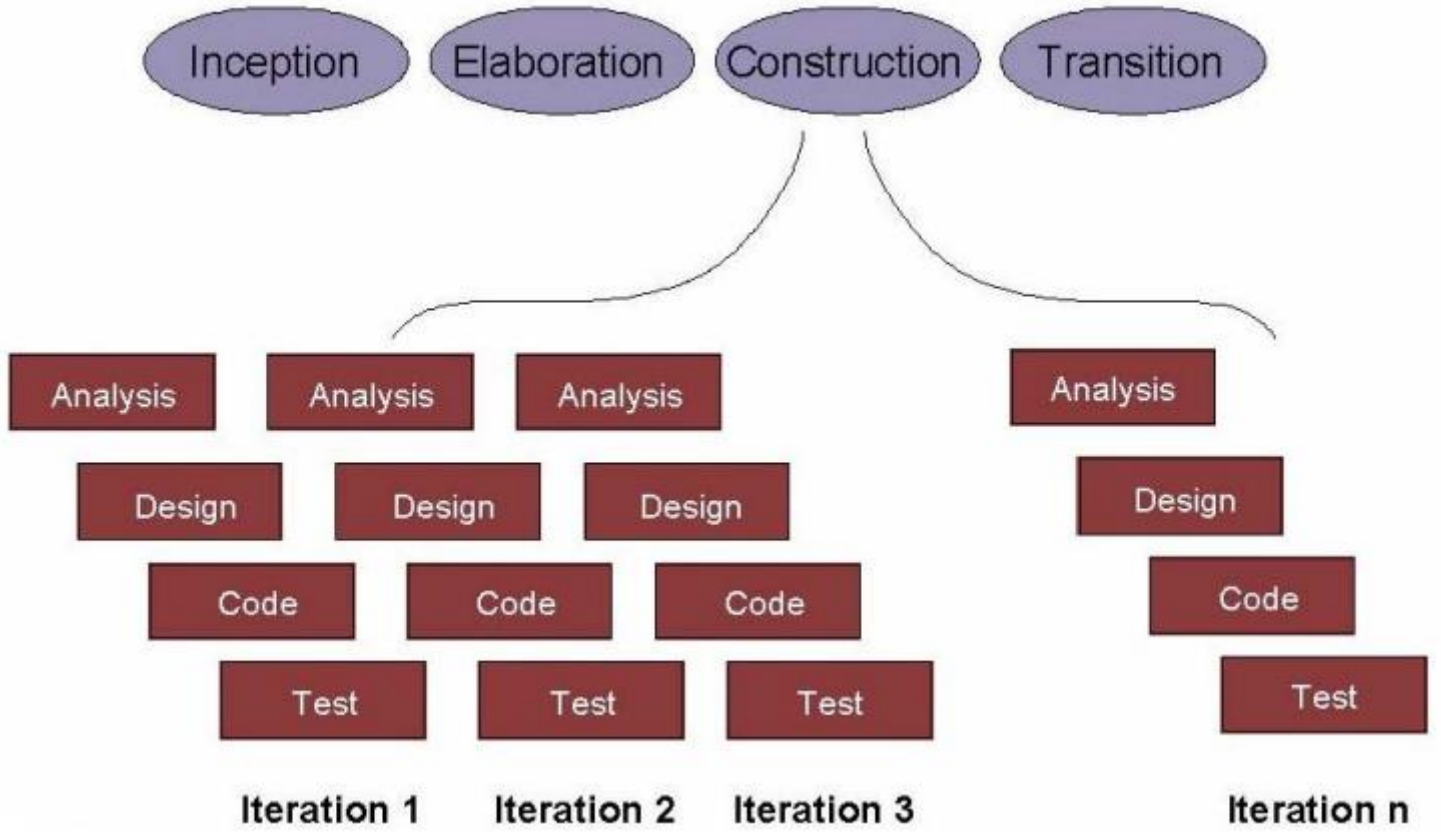
Other Plan & Document process models

- RUP
  - 1990
    - more closely allied to business issues than to technical issues
    - Like Waterfall and Spiral, RUP has phases
      plus
    - Six engineering disciplines



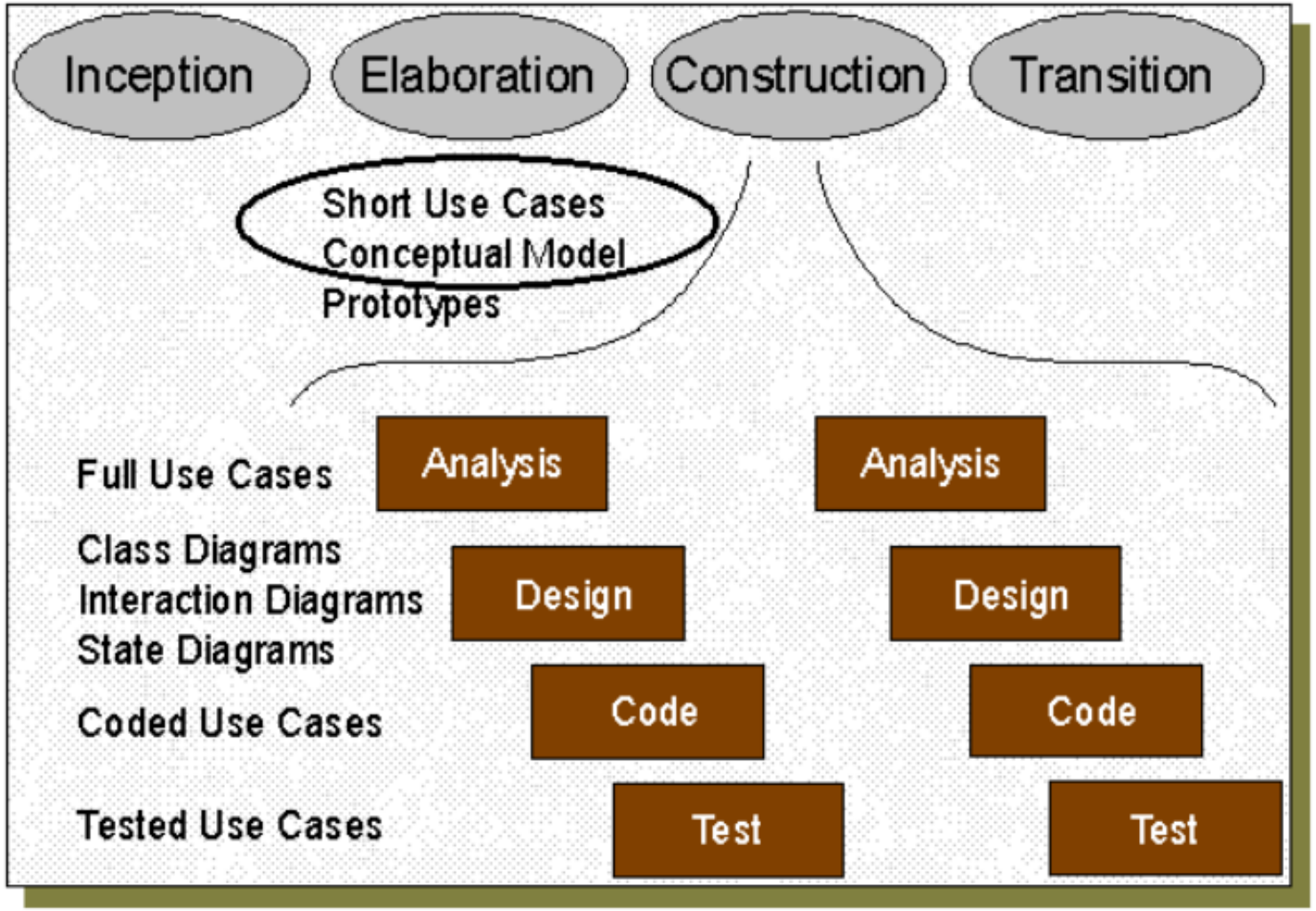
Other Plan & Document process models

- RUP
  - In action
    - Phase artifacts



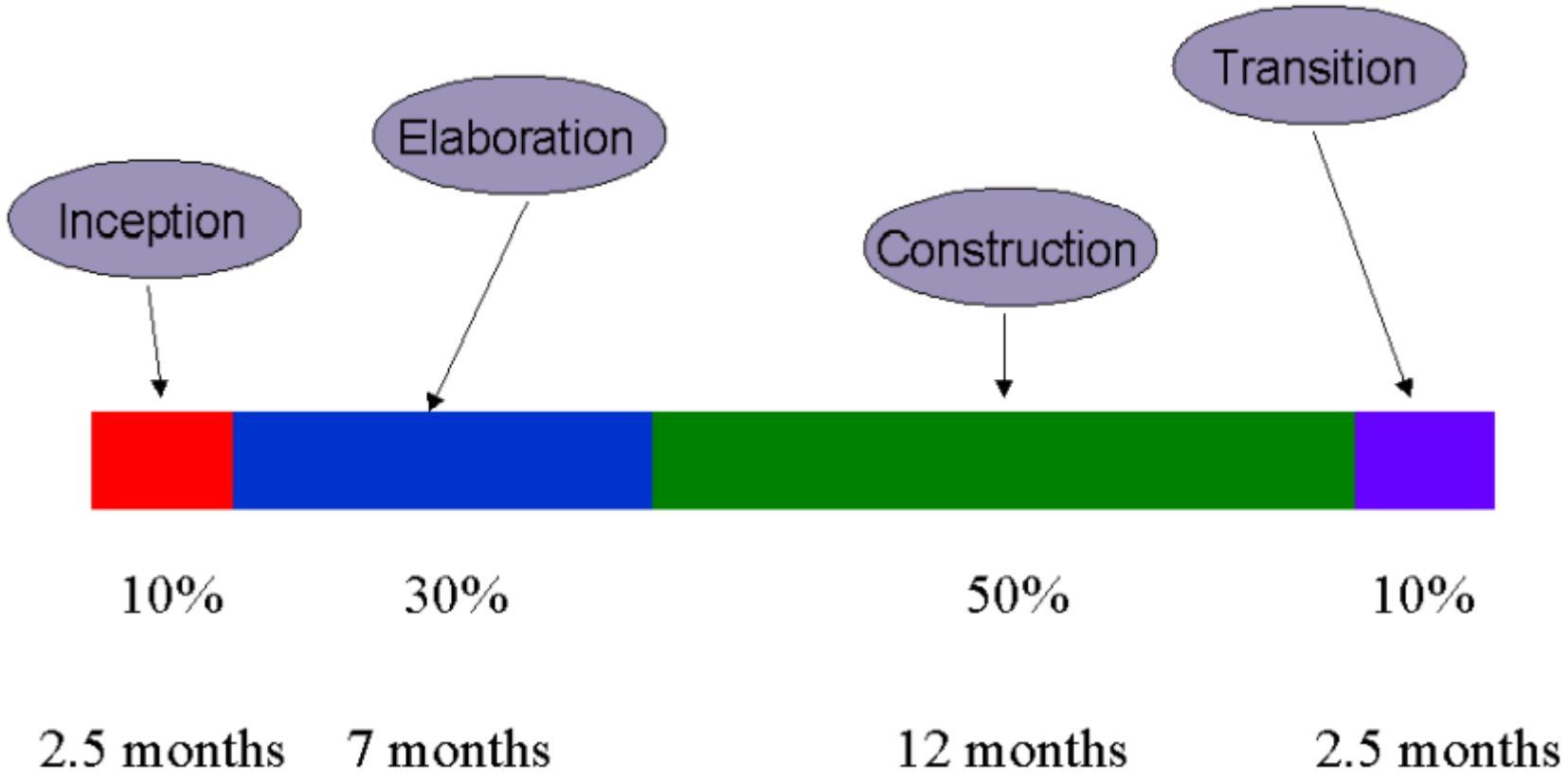
Other Plan & Document process models

- RUP
  - In action
    - Phase artifacts



Other Plan & Document process models

- RUP
  - In action
    - Timing

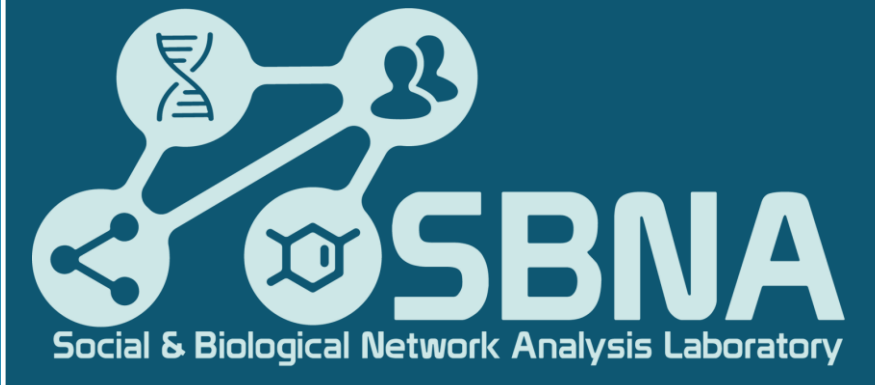


Downside to teaching a Plan-and-Document approach

- students may find software development tedious
- good news
  - is that there are alternatives that work just as well for many projects that are a better fit to the classroom, as we describe in the next section

Conclusion

- “The firms that typically get contracts are the firms that are good at getting contracts, not typically good at executing on them.”
- Another noted that the Plan-and-Document approach is not well suited to modern practices, especially when government contractors focus on maximizing profits



Question?

Bioinformation.ir

info@Bioinformation.ir