

ITP2200

Introduction to Software Testing

Bogdan Marculescu

Lecture 8

Test Processes, Test Plans

Software Development Process

Quick primer on how software development works:

1. Requirements analysis and specification
2. System and software design
3. Intermediate design
4. Detailed design
5. Implementation
6. Integration
7. System deployment
8. Operation and maintenance



Test Process



Quality

- At all the levels presented there!
- At each step – there are activities that can improve quality, and support or enable better testing
- It's not just code being assessed!

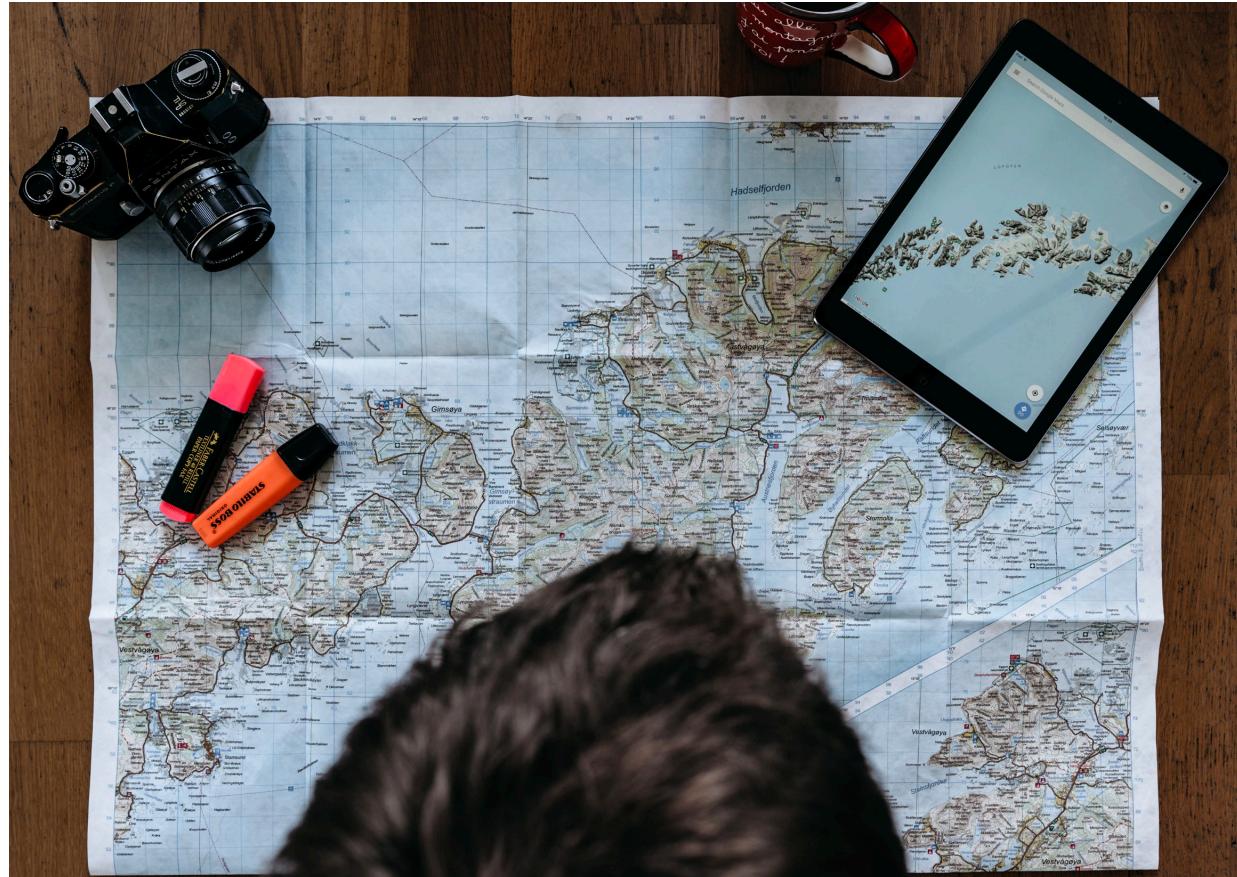


Høyskolen
Kristiania



Questions so far?

Previously on Introduction to Testing



We looked at writing tests in practice:

- Some regression testing
- Some integration testing

So, what can we conclude about our Narrator System?

Is anything missing?
Can we stop testing?

So what is missing?

I would argue, yes, many things ARE missing.

- Some methods are still not covered
- Do we cover all the partitions of the input space?
- How much more time do we need? (when can we release?)
- What happens if we release now?

We need a bit of order...



So what is missing?



We need... A PLAN!

The book has detailed information on test plans, including:

- Criteria for stopping,
- Tasks, responsibilities
- Deliverables (notice, these differentiate between the plan, the specifications, the process, the logs, reporting)
- Tools needed
- Risks and knowledge required



Høyskolen
Kristiania



Questions so far?

So what is missing?



We need... A PLAN!

- Motivation – hard to explain for artificial examples
- Tasks, responsibilities, choice motivations, tools, knowledge – essential for larger teams.
- Deliverables. Differentiate between specification:
 - Overview of what to test and link to requirements
 - Process – how to run the test (if manual) or test code (if automated)
 - Inputs – separate the inputs from the test code

So what is missing?

We need... A PLAN!

So, we don't build stuff fully from scratch:

- We can add more specifications, and high level tests
- We can add more test code (and we did)
- We can add more test data

All documents (test and otherwise) are living documents: can be changed and updated as needed.



So what is missing?



So, we have some tests for RQ1.

TODO 1. Display the time since an event (print "This many years/months/days ago").

Top down:

What tests do I need to assess this?

Preconditions?

Postconditions?

So what is missing?



TODO 2. Print the duration of an event (print "This event took [x] years/months/days") IF the event took more than a day

Top down:

What tests do I need to assess this?

Preconditions?

Postconditions?

So what is missing?



Now for RQ3.

TODO 3. Print the time that passed between the current event and the previous event (e.g. "It was [x] years/months/days later that [y] happened") IF the previous event happened before.

Top down:

What tests do I need to assess this?

Preconditions?

Postconditions?

So what is missing?



At this level:

- high-level, general tests
- For example, RQ2 my tests should include:
 - one date less than a month ago
 - One date less than a year ago
 - One date more than a year ago

So what is missing?

Bottom up:

- Ensure all the pieces are okay
- Integrate them together
- Integration tests
- Regression tests
- We did a bit of this, too



Software Development Process

Quick primer on how software development works:

1. Requirements analysis and specification
2. System and software design
3. Intermediate design
4. Detailed design
5. Implementation
6. Integration
7. System deployment
8. Operation and maintenance





Høyskolen
Kristiania



Questions so far?

Exercise for the Seminar



Online!

- Put together a **test plan** for our Narrator
- Focus on the motivation, link the RQs to a clear plan for testing
- Also have tests that are in line with your plan
- End with a section about what is missing, and what should be done next
- Send me the document for comments and feedback (not grades!)

