Software Engineering Project Workshop (SENG202)

Matthias Galster

Introduction to Phase 3

September 21, 2020

UC
UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

Shared space for this session

- Google Docs
  - https://docs.google.com/presentation/d/1PX2rvmsh184ugiB48qiMP6yfKWO9R8fpkODjsaH5B7k/edit?usp=sharing
- Link also on Learn
  - COVID-19 section under “Schedule changes” for 21 September
- Everybody can edit
  - No need to log in

Reminders

- Time left for Phase 2
  - 1 day
- Submit your weekly individual reflection (Mondays, 5:00pm)
- Keep logging as you go (follow instructions)
- Labs (for those not presenting Deliverable 2)
  - Same allocation of teams as last week
  - No tutorial/quiz: tutors will be available for help

Presentations of Deliverable 2

Reminder

- Wednesday: Teams 1, 3, 5, 7, 9; Thursday: Teams 2, 4, 6, 8, 10

- Presentations are to be recorded and played via Zoom
  - Record presentation (e.g., as recorded Zoom meeting)
  - Edit if needed (e.g., with OBS-Studio or OpenShot Video Editor)
  - Test that sharing video and sound works (if you are not host of meeting)
  - One team member will share video recording with class

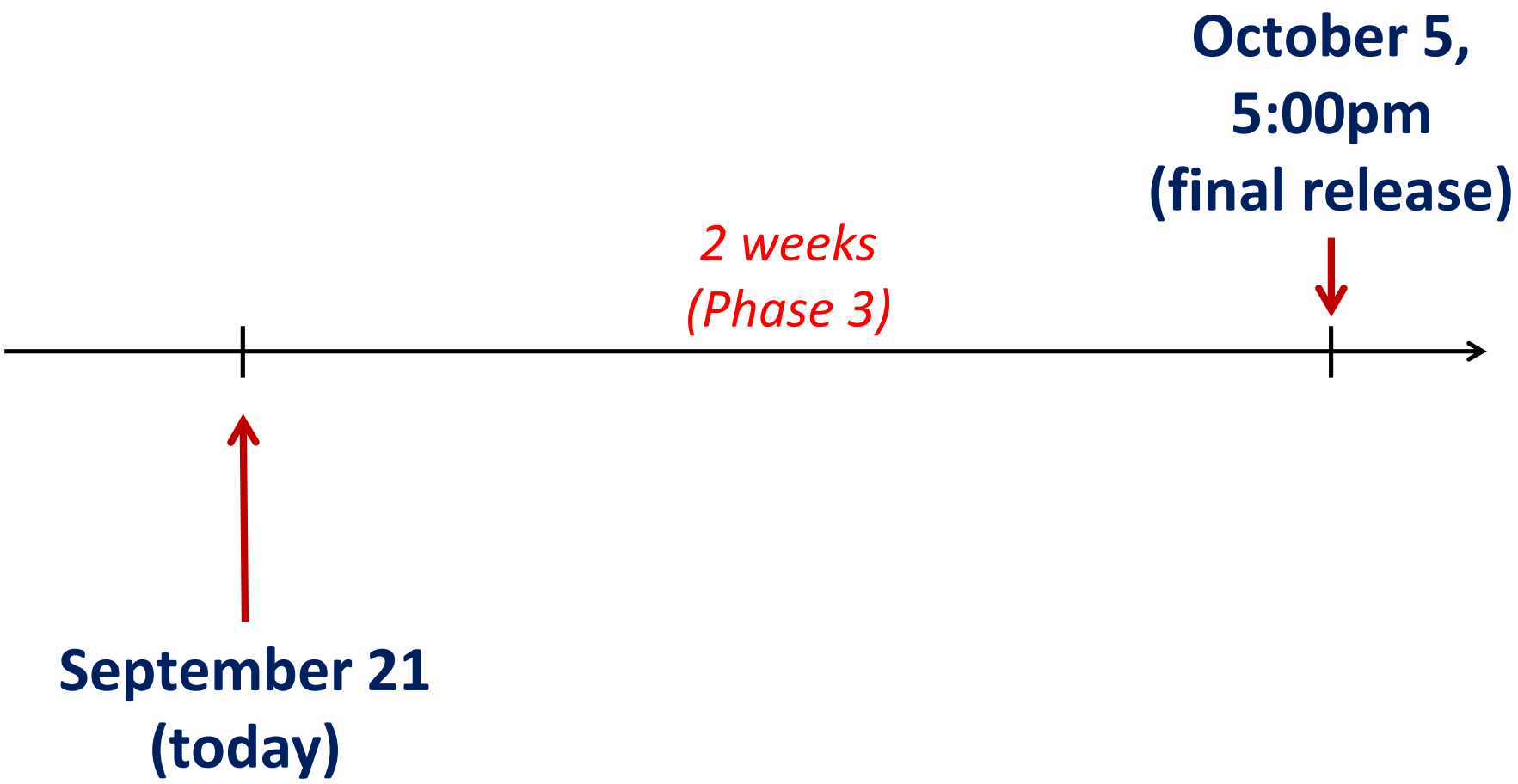
- Attend all presentations of session
  - Answer questions after own presentation
  - Ask questions after other presentations

- Content: see previous course notes

Last lecture

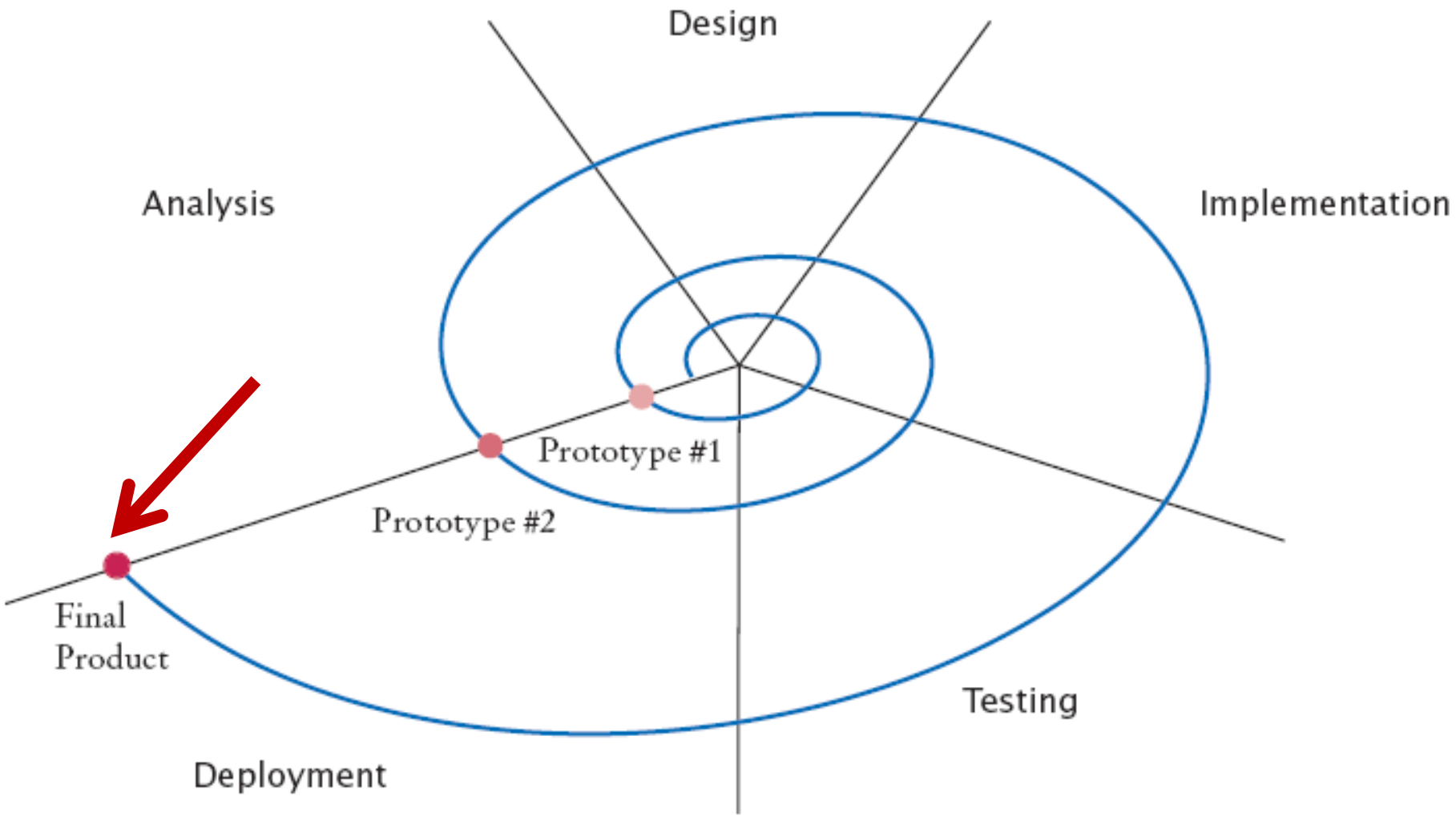
- What did we miss?
  - Examples of smells and what to do about them
- Reminders
  - What is a smell?
  - What do reviews have to do with smells?
  - What is refactoring?
- Examples?
  - Smells within classes, between classes?
  - How to find suitable refactorings (if no experience with smells)?

What is left…


October 5,
5:00pm
(final release)
2 weeks
(Phase 3)
September 21
(today)

Phase 3

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
September 21	September 22	September 23	September 24	September 25	September 26	September 27
September 28	September 29	September 30	October 1	October 2	October 3	October 3
October 5	October 6	October 7	October 8	October 9	October 10	October 11
October 12	October 13	October 14	October 15	October 16	October 17	October 18


Design
Analysis
Implementation
Prototype #1
Prototype #2
Final
Product
Testing
Deployment

Focus: Finishing and project delivery

Goals

- Produce final version of product

- Deliver product and project

Deliverables

- Complete and final design document
- Product
- Reflections and logs
- Demo to course staff

Deliverables

- Complete and final design document
  - Revised report from Phase 1 (see instructions for Phase 1)
  - Revised sections added for Phase 2 (see instructions for Phase 2)
  - **New (numbered) section at the end of the design document**
    - **Two to three key lessons learned (for each team member)**
- Product
- Logs
- Demo to course staff

Complete and final design document

Executive summary (update)

Change log (update)

1. Business and system context (update)
2. Stakeholders and requirements (update)
3. Acceptance tests (update)
4. GUI prototypes (update, but do not retrofit)
5. Deployment model (update)
6. Detailed UML class diagram (update)
7. Testing procedures (update)
8. Current product version (update)
9. Risk assessment (update)
10. Project plan (update)
11. Lessons learned (new)

Map to tasks

References (update)

Appendix (update)

Deliverables

- Complete and final design document

- Product (make sure it compiles, runs, can be imported)
  - See Phase 2
    - Exported project (from IDE) + executable product (as .jar)
    - Documented source code, Javadoc, JUnit tests, Cucumber tests
    - README (how to install/deploy/run app, how to import/build project, etc.)
    - Other resources required to run the app
  - **User manual as PDF (how to install/use app, etc.)**

- Reflections and logs

- Demo to course staff

Deliverables

- Complete and final design document
- Product
- Reflections and logs
  - As in Phase 2
- Demo to course staff

Deliverables

- Complete and final design document
- Product
- Logs
- Demo to course staff

Demo to course staff

- Presentation and demo to lecturers, tutors
  - 20 minutes in total (i.e., including questions)
  - 25% of Deliverable 3
  - Demo and questions about product, project, process, practices

- Contingency plan
  - Where
    - If Alert Level 1 or 2 (no physical distancing): Erskine 243
    - If Alert Level 2 (physical distancing): Erksine 243 or lecture/seminar rooms
    - If Alert Level 2, 3 or 4 (online): Zoom
  - If on Zoom
    - Demo must include recorded presentation
    - Be available for discussions

Presentation and demo

- Begin with overview of project, goals, stakeholders, use cases
- Then, show how product works and let us play with it
  - Reminder
    - Test your demo in the same environment as the presentation
    - It is very likely your demo will not work if you do not
  - Prepare and rehearse (some) scenario(s)
- Show some code and design (see next slide)
- Finally
  - Discuss technical challenges you overcame
  - Discuss lessons learnt

Code and design in the demo

- Prepare review of the most interesting sections of code
  - No need to show all code; do some of the critical, core and unique aspects
  - Give brief review of project context around each fragment
    - Code should not be totally out of the blue
  - Main authors/testers should lead that part
- Show related architecture, design (e.g., high-level UML models)
  - Use UML diagrams to better illuminate code, e.g., class diagrams
- Prepare for questions and to show additional code to answer

Plan and rehearse

- Make sure that the app works, set up demo before the demo
- Check timing (aim for presenting 10 to 12 minutes)
- Use of slides
  - Not many slides necessary
  - Might complement actual demo (maybe even use two screens)
  - Keep short, use for figures, code snippets (if not shown in IDE)
  - Rather than talking about features on slides, show features in application

What we are looking for in the demo

- Was the demo well-structured?
- Was a working product produced?
- How advanced is the project (features, quality, design)?
- Did errors or glitches arise? Exceptional cases handled gracefully?

But also

- Do team and team members have an understanding of
  - project,
  - process,
  - product,
  - practices?

Schedule

- October 5
  - 9:00am – 9:20am: Team ?
  - 9:25am – 9:45am: Team ?
- October 7
  - 10:00am –10:20am: Team ?
  - 10:25am – 10:45am: Team ?
  - 10:50am – 11:10am: Team ?
  - 11:15am –11:35am: Team ?
- October 8
  - 12:00pm – 12:20pm: Team ?
  - 12:25pm – 12:45pm: Team ?
  - 12:50pm – 1:10pm: Team ?
  - 1:15pm – 1:35pm: Team ?

How to submit

- Through Learn
  - By 5:00pm on the submission date
  - No drop-dead date

- As ZIP containing all deliverables
  - `seng202_2020_team<team>_phase<phase>.zip`
  - Must contain all the previously mentioned files and resources

Reminder

Start now

Ask for feedback early

Next steps

- Identify features for final release
- Identify other tasks
- Create plan for phase
- Code (write test for class, implement class), document, test, write

Hints

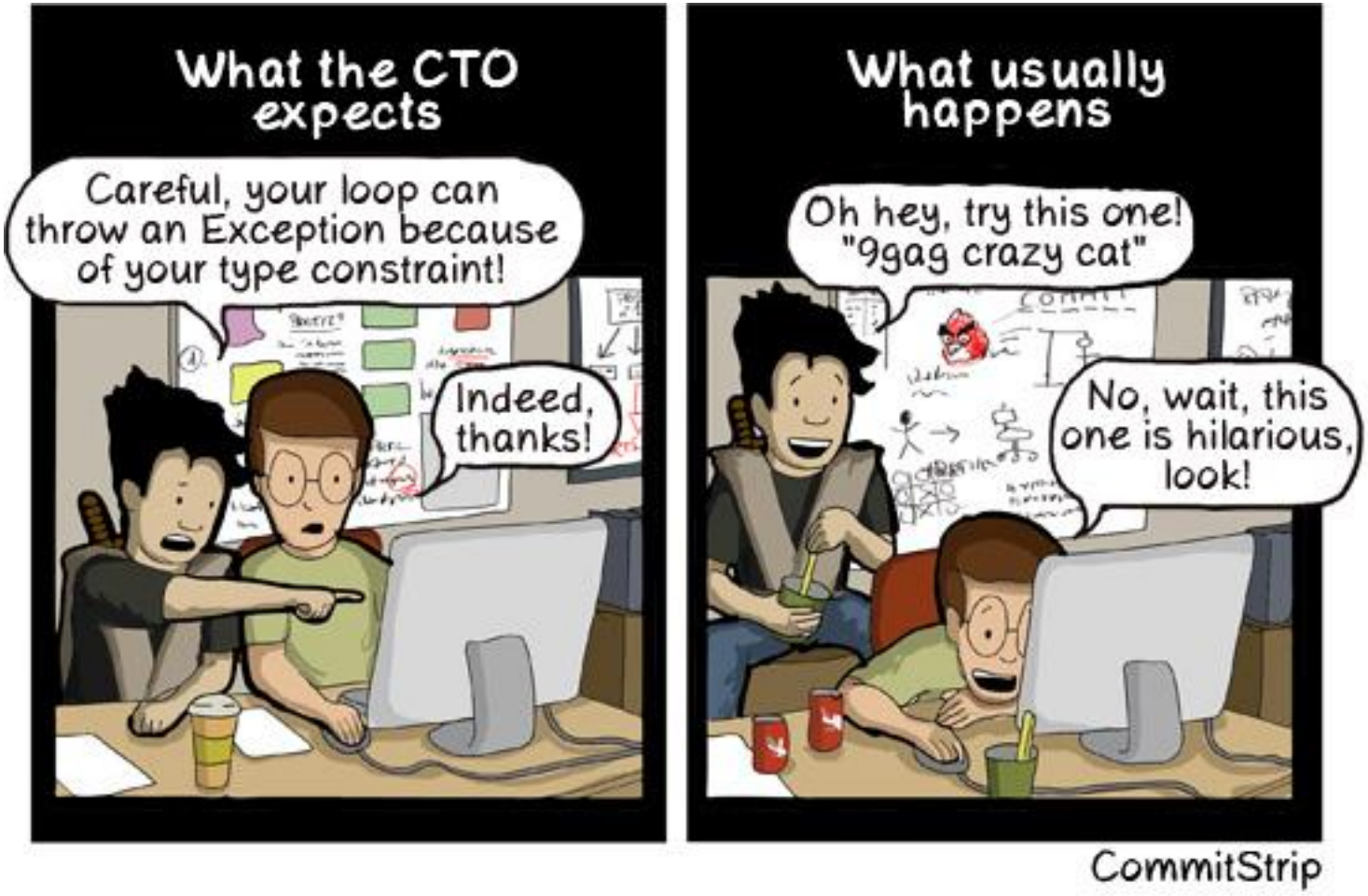
This phase is about wrapping up the project and preparing the final release. Thus, your product must be feature-complete and achieve the qualities and key drivers that you defined during analysis. The final product may include some additional features not included in the previous release. However, we recommend not adding any major features at this stage. Including major and complex features at this stage could result in additional problems and unanticipated challenges that may put this final phase and the final product at risk. You may want to discuss your plan for this last phase with teaching staff if you are not sure if you should include new features. Pay attention to the usability of your product, i.e., is the GUI intuitive to use, are invalid inputs treated properly, etc.

Make sure that your product can easily be installed and set up by customers. For example, if starting a server requires some configurations, you may create a batch file where users can set parameters, rather than merely providing a .jar file that is started from a command line. Also, you should create a brief user manual that describes how users would install and use the product and introduce the main features (screenshots usually help).

Final testing and documentation

Testing is essential to ensure our confidence in the quality of the product. The product must therefore once again be thoroughly tested with unit tests and acceptance tests. As in the previous phase, you are required to maintain a test protocol to be included in the final design document. The test protocol must also report the code coverage of your testing activities and list parts of your code with low test coverage. Furthermore, the test protocol must list any other problems that occurred during testing. You can update and maintain the description of testing procedures from the previous phase (see also instructions for Phase 2 for what to include in the discussion of the testing procedures).

When finalizing the documentation, you should make sure that the documentation is actually useful and check whether they would help other stakeholders and developers who want to understand and maintain your system.


What the CTO expects
Careful, your loop can throw an Exception because of your type constraint!
Indeed, thanks!
What usually happens
Oh hey, try this one! "9gag crazy cat"
No, wait, this one is hilarious, look!
CommitStrip