Software Engineering

Best Practices for Final Year Projects

by
Dr. Andy Chun, Hon Wai
Subject Group Leader for SE
Department of Computer Science
Content

- FYP Lifecycle
Content

- Agile Principles
  1. Talk to Your Advisor
  2. Work with Your Advisor
  3. Software Must Work
  4. Manage Changes
Content

- FYP Best Practices
- 5. Be an Expert
- 6. Manage Project Scope
- 7. Develop Iteratively
- 8. Test Your Software
- 9. Learn to Manage Yourself
- 10. Practice Makes Perfect
Remember!

- Your FYP is the single most important project in your whole undergraduate life!
FYP Lifecycle

Process Workflows
- Business Modeling
- Requirements
- Analysis & Design
- Implementation
- Test
- Deployment

Supporting Workflows
- Configuration Mgmt
- Project Management
- Environment

Phases
- Inception
- Elaboration
- Construction
- Transition

Iterations
- Preliminary Iteration(s)
- Iter. #1
- Iter. #2
- Iter. #n
- Iter. #n+1
- Iter. #n+2
- Iter. #m
- Iter. #m+1
4 Key Phases

1. Inception Phase
2. Elaboration Phase
3. Construction Phase
4. Transition Phase
Inception Phase (Sept)

- **Tasks:**
  - Finalize FYP with advisor
  - Do some basic reading to help decide
  - Set goals and objectives for yourself

- **Objectives:**
  - Common understanding between you and advisor on scope of work

- **Deliverables:**
  - “Project Plan” – a detailed proposal with project schedule
Elaboration Phase (Oct–Nov)

- **Tasks:**
  - Perform background research
  - Define system requirements
  - Based on research, design your system

- **Objectives:**
  - Detailed design based on solid research

- **Deliverables (may need 2 iterations):**
  - “Interim Report” (10%) – background research and initial design
Construction Phase (Dec–Mar)

- **Tasks:**
  - Coding and testing
  - Experimentation and Analysis
  - Demonstrate prototype to advisor for feedback

- **Objectives:**
  - Finish software development

- **Deliverables (may need 2 iterations):**
  - Source code with documentations
  - Demonstration to advisor and assessor
Transition Phase (Mar–Apr)

Tasks:
- Document work as Final Report
- Prepare for presentation

Objectives:
- Wrap up and present project

Deliverables:
- “Final Report” (30%)
- Source code
- “Project Presentation” (10%)
10 FYP Best Practices

- My own personal observations for the past 10 years at CityU
- Note:
  - This is only my own personal opinion
  - Follow your supervisor’s guidelines first
1. Talk to Your Advisor
Notes:

- Report what you are doing at least once a month
  - Preferably once every 2 weeks
- Tell advisor:
  - What papers you have read
  - What you have learned
  - Your design/coding progress
Software Engineering Principles:

- **Agile Manifesto:**
  - Individuals and interactions over processes and tools

- **XP Value:**
  - Communication
2. Work with Your Advisor
Notes:

- Discuss problems with your advisor
- Seek guidance and verification on project direction
- Give advisor drafts of all deliverables for feedback
- Understand advisor expectations
  - Understand minimum requirement for a “pass”
  - Understand what makes an “A” project
Software Engineering Principles:

- **Agile Manifesto:**
  - Customer collaboration over contract negotiation

- **XP Values:**
  - Feedback
  - Courage
3.
Software Must Work
Notes:

- Make sure your scope is not too large
- Divide your project into stages
  - At least the first stage must work
- Have a prototype working by end of Semester A to be safe
Software Engineering Principles:

- Agile Manifesto:
  - Working software over comprehensive documentation
4.

Manage Changes
Notes:

- Changes are unavoidable for any project
  - Learn how to control change

- But make sure changes do not impact schedule
  - If so, reduce or modify scope if needed
  - FYP title can be changed any time
  - Need to have prototype working asap to know if heading towards a dead-end
Software Engineering Principles:

- Agile Manifesto:
  - Responding to change over following a plan

- RUP Best Practice:
  - Control changes to software
5. Be an Expert
Notes:

- FYP is an opportunity for you to demonstrate you can learn on your own
- Show what you have learned by being an expert on your FYP topic
  - Know what current state-of-the-art
  - Know all players: researchers/products
  - Know current technology direction
  - Know success stories
- Technical merit is 40% of FYP
6. Manage Project Scope
Notes:

- Must have a checklist of requirements
  - Prioritize this list
  - Estimate effort/time for each item
  - Verify all requirements can be done within schedule
Software Engineering Principles:

- RUP Best Practices:
  - Manage requirements
- XP Best Practices:
  - Planning Game
7.

Develop Iteratively
Notes:

- You should have at least 2 or 3 iterations
  - Iteration 1 is prototype – used to get feedback from advisor
  - Iteration 2 is normal FYP deliverable
  - Iteration 3 is enhanced FYP for “A” student
Software Engineering Principles:

- **RUP Best Practices:**
  - Develop software iteratively

- **XP Best Practices:**
  - Small Releases
8.

Test Your Software
Notes:

- Must have at least unit test cases
  - Use *Unit
  - Use Log4*
- Do benchmarking and comparisons
- Must be able to compare:
  - Compare your results with results from other approaches for same or similar problems
Software Engineering Principles:

- RUP Best Practices:
  - Verify Software Quality
- XP Best Practices:
  - Test-driven
9.

Learn to Manage Yourself
Notes:

- Manage your time
  - FYP is equivalent of 2 courses per semester
  - You have only 13 + 9 weeks to do FYP
  - We expect you to put in roughly 300 hours total
    - Roughly 2 days/week (14 hours)
  - Write down what you have done each week
Notes:

- Keep track of:
  - Time spent each week
  - Pages of document written each week
  - LOC written each week
  - Bugs found each week
  - Requirements satisfied each week

- Compare with original schedule

- Project management skills count 10% of FYP
Software Engineering Principles:

- Personal Software Process (PSP)
  - From the CMM folks
10.

Practice Makes Perfect
Notes:

- Make sure you practice your presentation beforehand
  - Give advisor a copy of PPT at least 1 week before to get feedback
  - Dry run in front of friends/classmates
References:

- **Agile Manifesto**
  - agilemanifesto.org

- **RUP**

- **XP**
  - extremeprogramming.org

- **PSP**
  - www.sei.cmu.edu/tsp/psp.html
Most Important Best Practice

100.

Remember to Have Fun
Questions