Good software practices

Fethi Rabhi
What is a successful software project?

- Successful project can be measured by
  - Happy customers
  - Happy teams
  - Software that is being used
- Success does not come from a prescriptive process but experienced people who possess
  - Rules of thumb
  - Practices, not processes
- Learning from failures is encouraged
Processes vs practices

- Processes prevent the mediocre from making mistakes.
- Practices make the professional exceptional.
Good practice requires

<table>
<thead>
<tr>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project planning</td>
</tr>
<tr>
<td>Agile principles</td>
</tr>
<tr>
<td>User requirements management</td>
</tr>
<tr>
<td>Agile roles</td>
</tr>
<tr>
<td>Agile tools</td>
</tr>
</tbody>
</table>
Project Planning
It helps to know where you are going...

How do I plan and estimate?

- Understand the goals
- Plan work to make the goals happen
- Think about risk and how to minimise it
- Write it down and share it
- Estimation...a problem

Is it just an exercise in futility?

- Start with a broad brush and paint in detail as you go
- Invest the right amount in planning
Typical Methodologies

Fancy names for methodologies
- Waterfall, Iterative, Spiral, RAD

Use Agile principles and here's why...
- It's a natural process
- It brings risk forward, rather than pushing it to the end
- It puts quality, value and end-user priority first
Agile Principles
Project Shape – Small sprints

Image available at
www.mountaingoatsoftware.com/scrum
Waterfall - Agile

PLAN  DESIGN  BUILD  TEST  REVIEW  MAINTAIN
Waterfall - Agile

1. Add posting
2. View postings
3. Search for posting
4. User login
How do you get started?

1. Define Problem
2. Document Stories
3. Sprint Planning
4. Track sprint as it goes
5. Retrospective
User Requirements Management
Your product is currently set up as *Epics*

- You need to break them into stories, prioritise them and elaborate them
- Start by elaborating just the story titles
- Before your first sprint, fully elaborate the stories for that sprint (at least)
Story telling tools

- Story cards
- Simple tools: Spreadsheets, whiteboards
What is a user story?

FORM:
As <PERSONA> I want to be able to <ACTION> because <JUSTIFICATION>

EXAMPLE:
• As Kerry I want to be able to upload a photo to share with my friends and win prizes
Personas

DEFINITION: A user of your application

• EXAMPLE:
Kerry is an 18 year old girl and loves using her iPhone. She is an avid Facebook user and regularly shares pictures with her friends.

• WHAT TO PUT IN:
Relevant facts to the project that app that you are developing

• WHAT NOT TO PUT IN:
Unrelated facts, e.g. hair colour, geographic location, things she likes, etc.
What goes in a story?

Title: User upload photo
Epic: Sharing Photos

1. User Story:
   • As Kerry I want to be able to upload a photo to share with my friends and win prizes.

2. Deliverables:
   • Kerry can upload a photo once she is logged in.

3. Exceptions and Notes:
   • Kerry cannot upload a photo when she is not logged in.
   • Kerry can only upload JPEG, PNG or GIF.
   • Kerry should be given feedback during the upload process, e.g. upload bar.
   • Kerry should be able to upload a photo from her phone.
Sprint Planning

Which stories go into the sprint?

Do the stories make sense together?

How hard is this going to be to complete?

Who can work on what?
Tracking the Sprint
Daily Standups (online ?)

WHAT DID YOU DO YESTERDAY?

WHAT ARE YOU GOING TO DO TODAY?

IS THERE ANYTHING IN MY WAY?
Demo your work and get feedback from the team.

What worked well?

What didn’t work well? How can we improve things?

Do we have something that works?
Agile Roles
Roles

• Assign them, assign them wisely
• Of course, there are many more roles but you don't need them in this case.
• Roles: People is not 1:1. One person can have many roles, you can swap roles.
• You can not be the Dev-Tester for your own code - you can be a developer and a tester but only testing other people's code
• You can't review your own work
Product Owner

Is a business analyst

Consults with stakeholders and documents their input

Creates stories that can be implemented
  - Does enough analysis up front
  - Sufficiently described
  - Are not designs

Works with QA and stakeholders to ensure acceptance criteria are right

Owns the product backlog

Prioritises all stories into backlog in consultation with the project sponsor
Scrum Master

Ensures engineers have no roadblocks and are doing what they should be doing

Keeps the PM updated re progress, risks, issues

Has daily meetings with her team

Provides technical leadership, design guidance, design review and code review

Ensures quality is maintained at an appropriate level

Helps his team to plan and estimate work

Updates Sprint Board Daily

Maintains backlog for the project, Updates backlog each Sprint

Facilitates Showcase and Retrospectives each Sprint
Developer

- Understands the story and acceptance criteria
- Analyses problems
- Designs solutions
  - in collaboration with architect, infrastructure etc
  - let's us know if major re-design is needed on existing systems
- Plans and Estimates their own work
- Codes and Unit Tests
- Participates in design and code reviews
- Asks for review and assistance
- Tests their own work against acceptance criteria
Developmental Tester

- Understands requirements provided by PO and writes acceptance criteria
- Develops & maintains test scripts
- Updates API test harnesses
- Ensures unit testing happens
- Ensures Dev Testing happens
- Reports on progress, raises bugs, retests
- Regression tests
- Monitors automated build and test
- Monitors code coverage
- Breaks stuff
Project Manager

- Is a Communicator, Facilitator and Trouble Shooter
- Understands project goals and benefits and shares vision with the team
- Ensures work is prioritised and scheduled
- Ensures work is planned by the engineers and is carried out as scheduled
- Manages project risks and ensures mitigation is executed
- Helps make sure project stakeholders are communicated with and that goals are met
- Maintains schedules and other project documentation
Team Activity 1: Each team completes the first 3 columns

<table>
<thead>
<tr>
<th>Team Member Name</th>
<th>GitHub email</th>
<th>Experience</th>
<th>Possible Roles (leave blank)</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Snow</td>
<td><a href="mailto:j.snow@xyz.edu">j.snow@xyz.edu</a></td>
<td>Programming in C++, Work experience in Banking</td>
<td></td>
</tr>
<tr>
<td>Patricia Pineo</td>
<td><a href="mailto:ppineo@Netscape.com">ppineo@Netscape.com</a></td>
<td>HTML/CSS, Graphic Design</td>
<td></td>
</tr>
<tr>
<td>Mark Hassan</td>
<td><a href="mailto:mmh@gmail.com">mmh@gmail.com</a></td>
<td>Hardware circuit design, Assembler coding</td>
<td></td>
</tr>
<tr>
<td>Alicia Key</td>
<td><a href="mailto:alkey@unsw.edu">alkey@unsw.edu</a></td>
<td>Python programming, Sales job experience</td>
<td></td>
</tr>
<tr>
<td>Alva Hyunh</td>
<td><a href="mailto:alva@toto.com">alva@toto.com</a></td>
<td>Double major in business, sports coaching</td>
<td></td>
</tr>
</tbody>
</table>
Agile Practices
Practices

- **Review** (documents, code, design, architecture, test scripts, everything)
- **Architecture** - have one
- **Design** - everything
- **Unit Testing** - no excuses
- **Regression Testing** - automated
- **Configuration Management** (software & environments)
- **Daily Check-in**
Practices

- Automated Testing
- Automated Build
- Continuous Integration - at least daily
- Daily meetings...
- Sprints (Momentum, Planning, Showcases, Retrospectives)
- Co-locate during development
Choosing collaboration tools

- Source control
- Meetings and discussions
- Project Management
- Document sharing
Activity 2: Each team must choose collaboration tools

- Populate remaining rows in table
  - Meetings and discussions e.g. WeChat, Facebook Messenger, MSTeams, Zoom, Discord
  - Project Management e.g. Trello, Asana
  - Document sharing e.g. Google Drive
- Same tools can be used in multiple rows
- Create more rows if necessary

<table>
<thead>
<tr>
<th>Collaboration Tool Type</th>
<th>Collaboration Tool Name</th>
<th>Account Manager</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source control</td>
<td>GitHub</td>
<td>Alva Hyunh</td>
<td>Repo address is...</td>
</tr>
<tr>
<td>Meetings and discussions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Management</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document sharing</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Agile Methodologies vs Agile Principles

They focus on:
- THINK
- INTERACT
To achieve agility