Agile Samurai Principles

The Agile Samurai
How Agile Masters Deliver Great Software

Agile Development

Three simple truths

1. It is impossible to gather all the requirements at the beginning of a project.

2. Whatever requirements you do gather are guaranteed to change.

3. There will always be more to do than time and money will allow.
Deliver Value Every Iteration

- Break big problems into smaller ones
- Focus on most important issues
- Deliver something that works
- Lots of customer feedback
- Change course when necessary
- You are accountable

Agile Planning
Agile Planning

Hold on ... You are making this sound WAY too easy.

What if it's a fixed bid contract and everything has to be done or we are all going to die ???

Agile Planning

Unrealistic plan Miracle Working software
Agile Lifecycle

Agile Team

- Blurred instead of fixed roles
- Characteristics of successful teams
  - Co-located, at least for initial meetings
  - Engaged customer
  - Self-organizing instead of top-down
  - Accountable and empowered
  - Cross-functional
What if I don’t have an engaged customer?

• Build credibility
  – Find a problem and make it go away
  – Show you are a fierce executor that will get things done and can help them
  – Might take a few iterations but they will see your value

The Agile Customer

Customer involvement $\Rightarrow$ Project success

I WOULD BE HAPPY TO ANSWER THAT.

“THE SOURCE OF TRUTH”

DECREASES WHAT GETS BUILT

MAKES THE TOUGHS TRADE-OFFS AROUND SCOPE

Sets the priorities

Most important

Out of scope
Agile Development Team

- Business analysts
- Project managers
- Technical writers
- Database administrators
- Programmers
- Testers
- UX designers
- Everyone else!

Agile Analyst

- Helps write user stories
  - I know what I want, but how do I describe it?
- Does detailed analysis
  - I sweat the details.
- Makes sure we've done our homework
  - Analysis artifacts

You can count on me to do our homework, for each and every iteration!
Agile Programmer

Think of me as a customer with a keyboard!

"Because the rubber hits the road when you code."

Agile Tester

I'll tell you if it's working. I'll also tell you if it's not.

"Because finding out in production isn't an option."

Tools/Architecture
Design
Development Practices

Acceptance criteria

Acceptance criteria

Tests for upcoming stories

Create permit

Acceptance criteria

Basic search

1 pt
3 pts
5 pts

Makes technical decisions

Exploratory
Stress
Load
Integration
Security

Estimates (with rest of team)

if X then Y;
Agile Manager

Hi! What can I do for ya?

Tracks how we are doing

Communicates the state of the project

Removes roadblocks standing in the team's way

“Watching the bottom line.”

Agile Usability Designer

I ❤️ customers

Uses a collection of tools and techniques to help create a compelling user experience

Overlaps with analysis

“Because it's cool to think about the customer.”
Kicking off a project

• The Inception Deck
  – Ten questions you’d be crazy not to ask before starting any software project
  – Gets everyone pointing in the same direction
    • Shared goals, vision, context

Inception Deck

• Collectively fill out a slide on to get a pretty good idea about what the project is, what it isn’t, and what it’s going to take to deliver
• Need to get customer/stakeholders involved
• It’s a living document
<Your project name>

<Your sponsors>

Why are we here?

• Important reason #1
• Important reason #2
• Important reason #3

<#1 reason for doing this project>
The elevator pitch

• For [target customer]
• who [statement of need or opportunity]
• the [project name]
• is a [product category]
• that [key benefit, compelling reason to buy].
• Unlike [primary competitive alternative]
• our project [statement of primary differentiation].

Product box

<product name>

fun picture

<slogan>
<benefit #1>
<benefit #2>
<benefit #3>
### The NOT list

| IN   | OUT |!
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### UNRESOLVED

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Your project community

- **<community#3>**
- **<team#2>**
- **Your core team**
- **Everyone else!**

... is always bigger than you think!
Technical solution

Technologies:
- <language>
- <libraries>
- <tools>
- <technology>

Danger!

Out of scope

What keeps us up at night

• <scary thing #1>
• <scary thing #2>
• <scary thing #3>
Don’t overdo it

The A-Team

<table>
<thead>
<tr>
<th>#</th>
<th>Role</th>
<th>Competencies/Expectations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Analyst</td>
<td>Comfortable with just-in-time analysis. Likes to test. Comfortable with rapid iterative development.</td>
</tr>
<tr>
<td>2</td>
<td>Developers</td>
<td>C#, MVC.NET, jQuery, SQL Unit testing, refactoring, TDD, continuous integration</td>
</tr>
<tr>
<td>0.5</td>
<td>Project manager</td>
<td>Responsible for outward facing communication Status reports, scope, budget, and reporting upwards</td>
</tr>
</tbody>
</table>
How big is this thing?

Ship it!

Construction ~3months
UAT 1 wk
Training 1 wk

This is a guess. Not a commitment.

Risk vs. Time

The risk of project failure increases over time – think small
The Test

1. Which of these forces is most precious to a software project?
   - a) Quality.
   - b) Time.
   - c) Scope.
   - d) Budget.

2. When faced with too much to do and not enough time, is it better to do the following:
   - a) Cut scope
   - b) Add more people to the project
   - c) Push out the release date
   - d) Sacrifice quality

3. Which is most painful?
   - a) Walking on fire
   - b) Chewing broken glass
   - c) Doing the Macarena
   - d) Asking your sponsor for more money
Trade-off sliders

<table>
<thead>
<tr>
<th>The classic four</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON&lt;-&gt;OFF</td>
</tr>
<tr>
<td>Feature completeness (scope)</td>
</tr>
<tr>
<td>ON&lt;-&gt;OFF</td>
</tr>
<tr>
<td>Stay within budget (budget)</td>
</tr>
<tr>
<td>ON&lt;-&gt;OFF</td>
</tr>
<tr>
<td>Deliver project on time (time)</td>
</tr>
<tr>
<td>ON&lt;-&gt;OFF</td>
</tr>
<tr>
<td>High quality, low defects (quality)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other important things</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON&lt;-&gt;OFF</td>
</tr>
<tr>
<td>Ease of use</td>
</tr>
<tr>
<td>ON&lt;-&gt;OFF</td>
</tr>
<tr>
<td>Community of users</td>
</tr>
<tr>
<td>ON&lt;-&gt;OFF</td>
</tr>
<tr>
<td>Detailed audits (log everything)</td>
</tr>
<tr>
<td>ON&lt;-&gt;OFF</td>
</tr>
<tr>
<td>&lt;insert yours&gt;</td>
</tr>
</tbody>
</table>

The first release

Ship it!

Construction: ~3 months
UAT: 1 wk
Training: 1 wk

3 people, 3 ½ months, $250K