BUILDING A SMARTER WEB

The Agile Drupalist

Methodologies & Techniques for Running Effective Drupal Projects

By Adrian “AJ” Jones (Canuckaholic)

Agenda

What We Will be Talking About Today...

- ✓ Introductions
- ✓ What kind of processes are out there?
- ✓ The “Iron Triangle” of project management
- ✓ What does it mean to be “Agile”?
- ✓ Why be Agile?
- ✓ The Agile Drupalist
- ✓ Mapping the Project Life Cycle
- ✓ Case Study

Introductions

A Little Bit About Me...

User Experience Designer. Business Analyst. Content Strategist. Information Architect. Interaction Designer.

Passionate about making better digital **experiences**.

My Experience in Agile:

- ✓ Certified Scrum Product Owner
- ✓ Have worked on large-scale, big-budget Agile projects:
  - ✓ **Kobo Instant Reader HTML5 App**
  - ✓ **New York State Education Department**
  - ✓ **Stanford University School of Engineering**
- ✓ **Masters** degree in Interactive Arts & Technology, where I was first introduced to the Agile philosophy.

Introductions

A Little Bit About YOU!

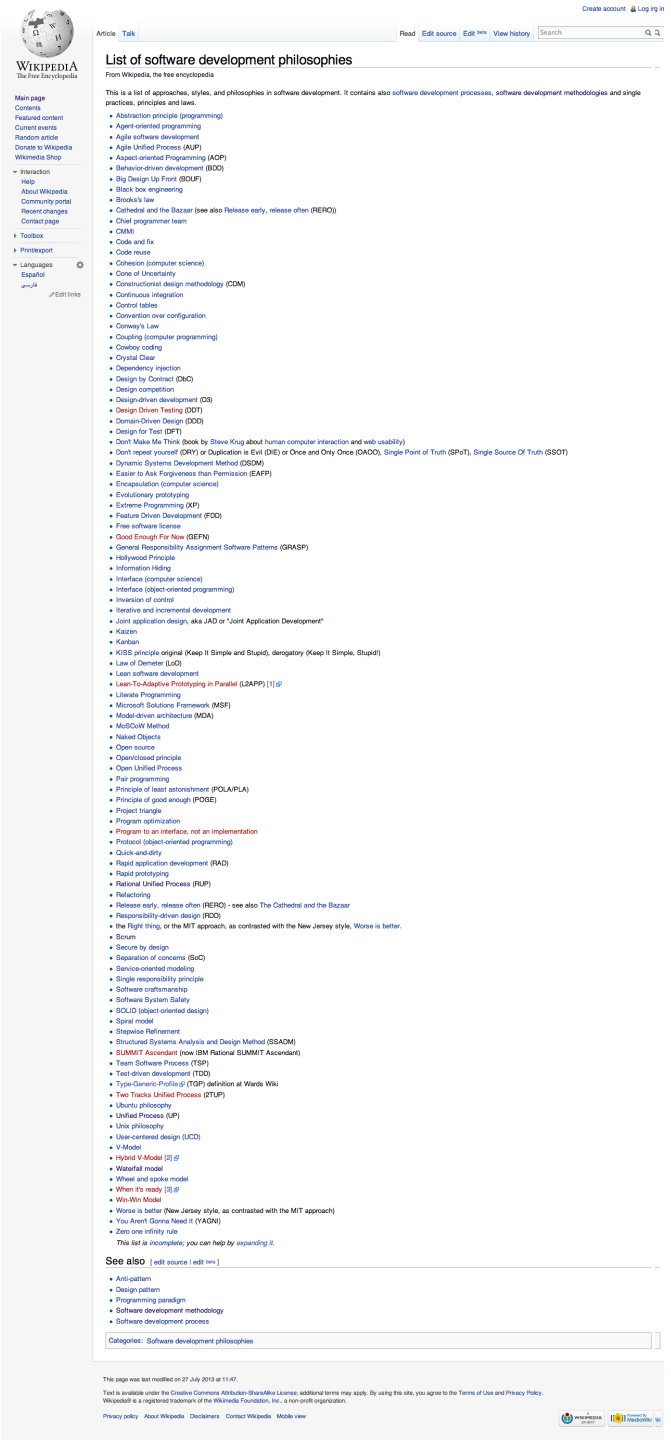
Who Are You?

- ✓ Web Dev?
- ✓ Front-end Dev?
- ✓ Back-end Dev?
- ✓ Drupal Dev?
- ✓ Web Designer?
- ✓ Information Architect?
- ✓ Business Analyst?
- ✓ UX Designer?
- ✓ Project Manager?
- ✓ All of the above?

What's Out There?

The Answer? A lot.

- ✓ Waterfall
- ✓ Agile
  - ✓ Scrum
  - ✓ Kanban
  - ✓ XP
  - ✓ Lean
- ✓ Rational Unified Process (RUP)
- ✓ Microsoft Solutions Framework (MSF)
- ✓ Rapid Application Development (RAD)



Create account Log in

Article Talk

Read Edit source Edit beta View history Search

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikimedia Shop

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Toolbox

Print/export

Languages
Español
فارسی
Edit links

List of software development philosophies

From Wikipedia, the free encyclopedia

This is a list of approaches, styles, and philosophies in software development. It contains also software development processes, software development methodologies and single practices, principles and laws.

- Abstraction principle (programming)
- Agent-oriented programming
- Agile software development
- Agile Unified Process (AUP)
- Aspect-oriented Programming (AOP)
- Behavior-driven development (BDD)
- Big Design Up Front (BDUF)
- Black box engineering
- Brooks's law
- Cathedral and the Bazaar (see also Release early, release often (RERO))
- Chief programmer team
- CMMI
- Code and fix
- Code reuse
- Cohesion (computer science)
- Cone of Uncertainty
- Constructionist design methodology (CDM)
- Continuous integration
- Control tables
- Convention over configuration
- Conway's Law
- Coupling (computer programming)
- Cowboy coding
- Crystal Clear
- Dependency injection
- Design by Contract (DbC)
- Design competition
- Design-driven development (D3)
- Design Driven Testing (DDT)
- Domain-Driven Design (DDD)
- Design for Test (DFT)
- Don't Make Me Think (book by Steve Krug about human computer interaction and web usability)
- Don't repeat yourself (DRY) or Duplication is Evil (DIE) or Once and Only Once (OAOO), Single Point of Truth (SPoT), Single Source Of Truth (SSOT)
- Dynamic Systems Development Method (DSDM)
- Easier to Ask Forgiveness than Permission (EAFP)
- Encapsulation (computer science)
- Evolutionary prototyping
- Extreme Programming (XP)
- Feature Driven Development (FDD)
- Free software license
- Good Enough For Now (GEFN)
- General Responsibility Assignment Software Patterns (GRASP)
- Hollywood Principle
- Information Hiding
- Interface (computer science)
- Interface (object-oriented programming)
- Inversion of control
- Iterative and incremental development
- Joint application design, aka JAD or "Joint Application Development"
- Kaizen
- Kanban
- KISS principle original (Keep It Simple and Stupid), derogatory (Keep It Simple, Stupid!)
- Law of Demeter (LoD)
- Lean software development
- Lean-To-Adaptive Prototyping in Parallel (L2APP) [1]
- Literate Programming
- Microsoft Solutions Framework (MSF)
- Model-driven architecture (MDA)
- MoSCoW Method
- Naked Objects
- Open source
- Open/closed principle
- Open Unified Process
- Pair programming
- Principle of least astonishment (POLA/PLA)
- Principle of good enough (POGE)
- Project triangle
- Program optimization
- Program to an interface, not an implementation
- Protocol (object-oriented programming)
- Quick-and-dirty
- Rapid application development (RAD)
- Rapid prototyping
- Rational Unified Process (RUP)
- Refactoring
- Release early, release often (RERO) - see also The Cathedral and the Bazaar
- Responsibility-driven design (RDD)
- the Right thing, or the MIT approach, as contrasted with the New Jersey style, Worse is better.
- Scrum
- Secure by design
- Separation of concerns (SoC)
- Service-oriented modeling
- Single responsibility principle
- Software craftsmanship
- Software System Safety
- SOLID (object-oriented design)
- Spiral model
- Stepwise Refinement
- Structured Systems Analysis and Design Method (SSADM)
- SUMMIT Ascendant (now IBM Rational SUMMIT Ascendant)
- Team Software Process (TSP)
- Test-driven development (TDD)
- Type-Generic-Profile (TGP) definition at Wards Wiki
- Two Tracks Unified Process (2TUP)
- Ubuntu philosophy
- Unified Process (UP)
- Unix philosophy
- User-centered design (UCD)
- V-Model
- Hybrid V-Model [2]
- Waterfall model
- Wheel and spoke model
- When it's ready [3]
- Win-Win Model
- Worse is better (New Jersey style, as contrasted with the MIT approach)
- You Aren't Gonna Need It (YAGNI)
- Zero one infinity rule

This list is incomplete; you can help by expanding it.

See also [edit source | edit beta]

- Anti-pattern
- Design pattern
- Programming paradigm
- Software development methodology
- Software development process

Categories: Software development philosophies

This page was last modified on 27 July 2013 at 11:47.

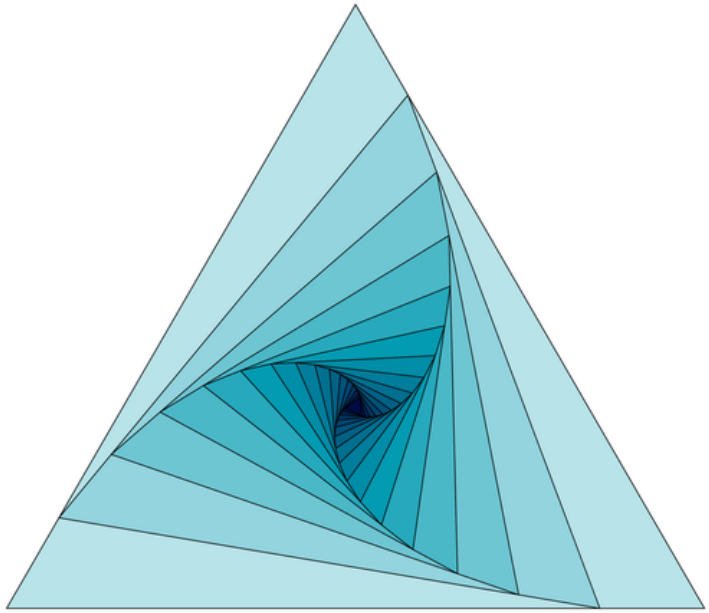
Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.

Privacy policy About Wikipedia Disclaimers Contact Wikipedia Mobile view

The Iron Triangle

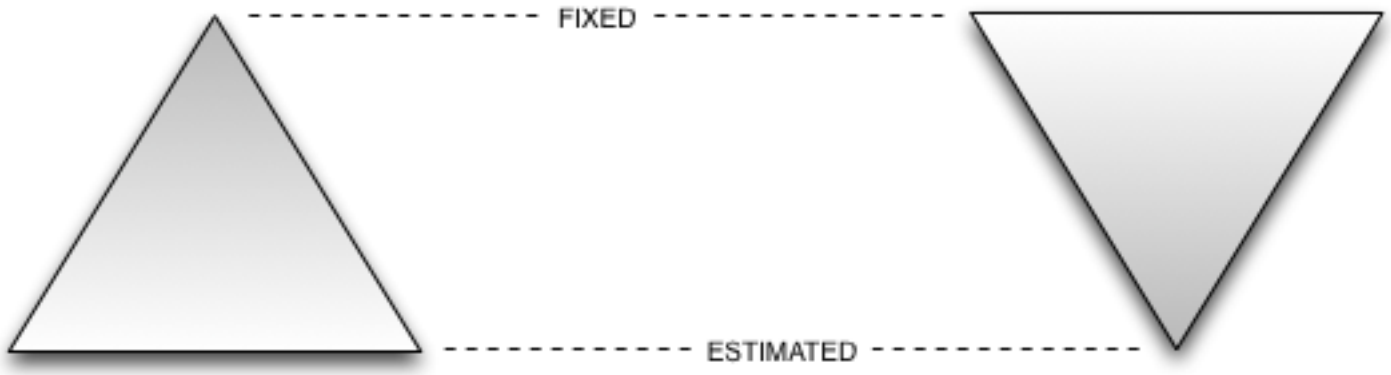
Of Project Management

Okay… Let's see how much we know…

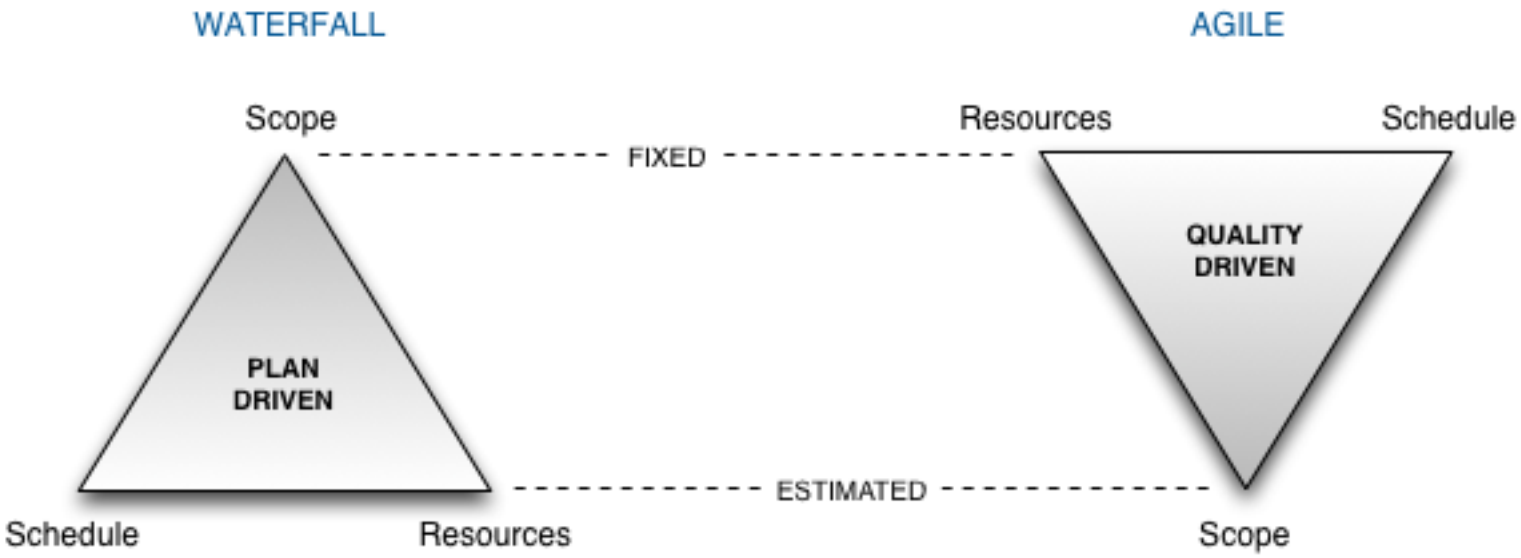


The 3 Points of the Iron Triangle Are:

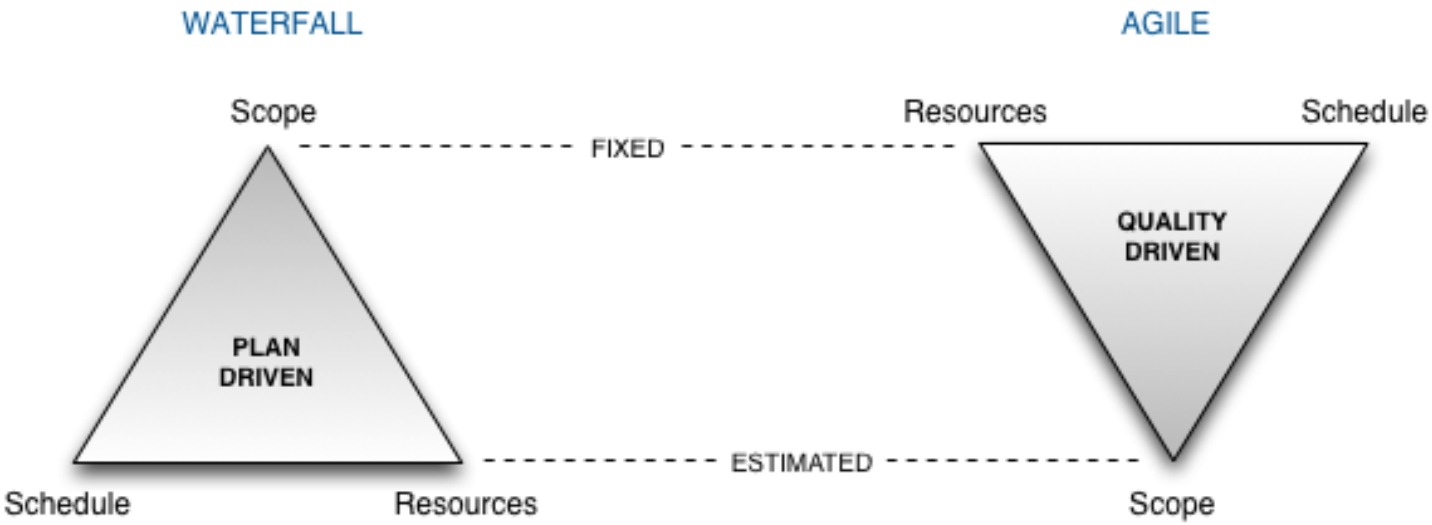
- ✓ Scope/Features/Functionality
- ✓ Budget/Resources
- ✓ Time/Schedule

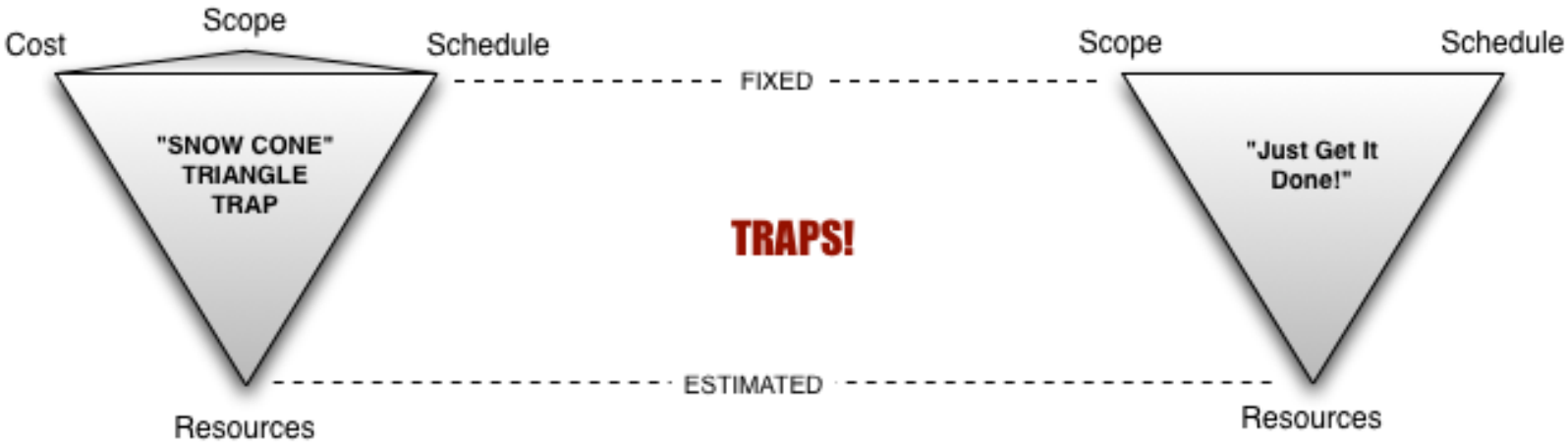


How I Would Draw It...



Some Traps


WATERFALL
AGILE
Scope
Resources
Schedule
FIXED
PLAN DRIVEN
QUALITY DRIVEN
ESTIMATED
Schedule
Resources
Scope


Cost
Scope
Schedule
FIXED
Scope
Schedule
"SNOW CONE" TRIANGLE TRAP
"Just Get It Done!"
TRAPS!
ESTIMATED
Resources
Resources

What is “Agile”?

Agile is NOT a Methodology… no really.

Agile (capital “A”) is a philosophy. It’s a way of thinking.

“Agile software development is a **group** of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.“ (Wikipedia)

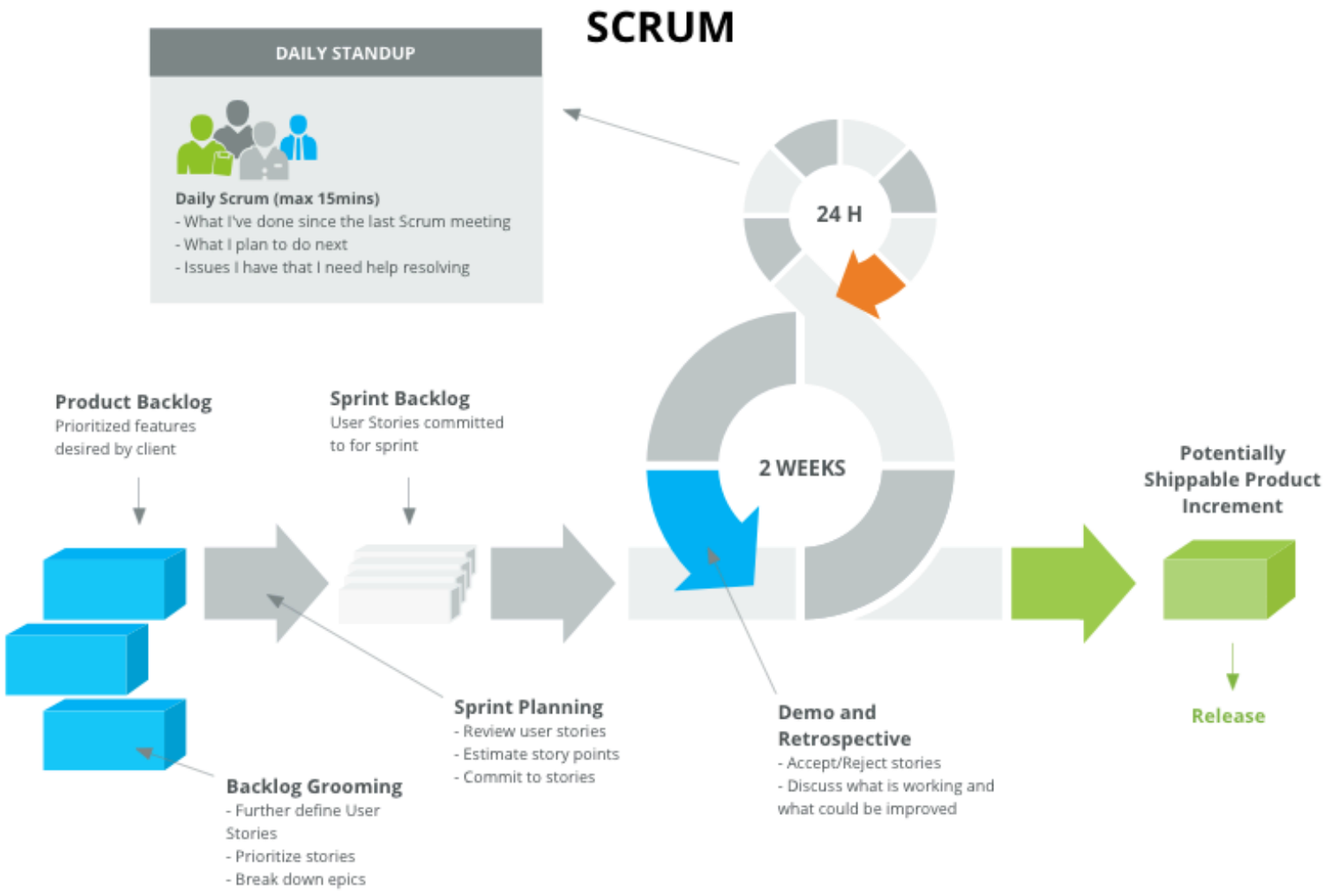
Scrum is an example of a ***methodology*** under the Agile ***philosophy*** of software development.

The Agile Manifesto

Consists of Only 4 Principles...

The Scrum Process

Standup. Groom. Plan. Demo. Release.



Why be Agile?

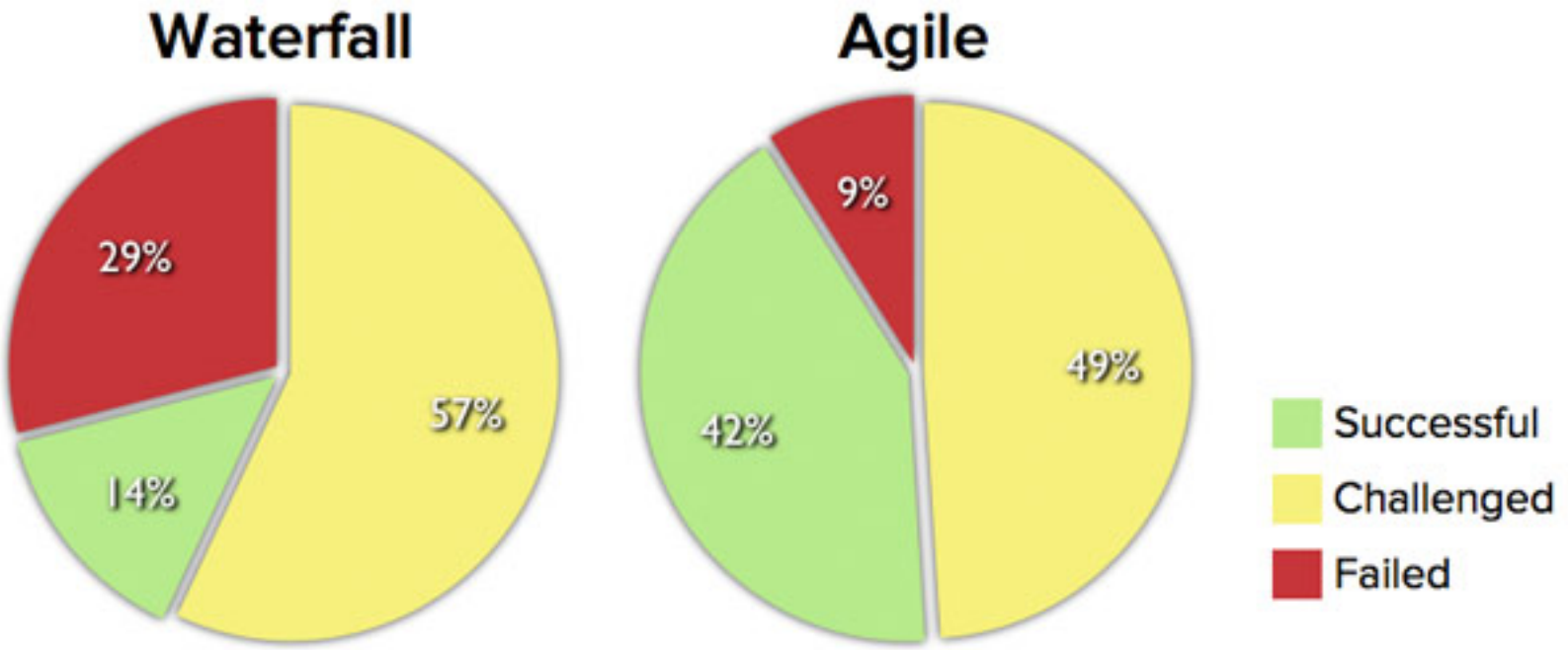
What are the benefits and why should I bother?

Because...

Why be Agile?

What are the benefits and why should I bother?

Because research shows that you have a greater chance of success with Agile methods.



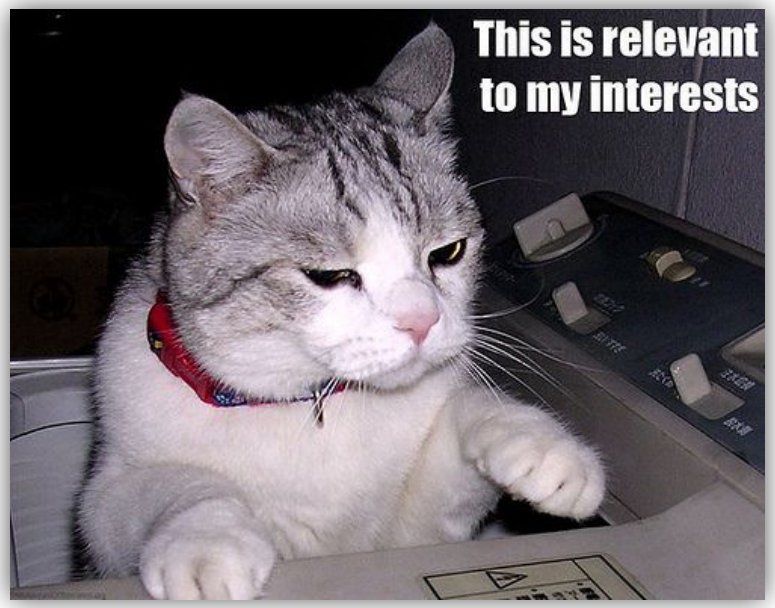
Source: The CHAOS Manifesto, The Standish Group, 2012.

Why be Agile?

What are the benefits and why should I bother?

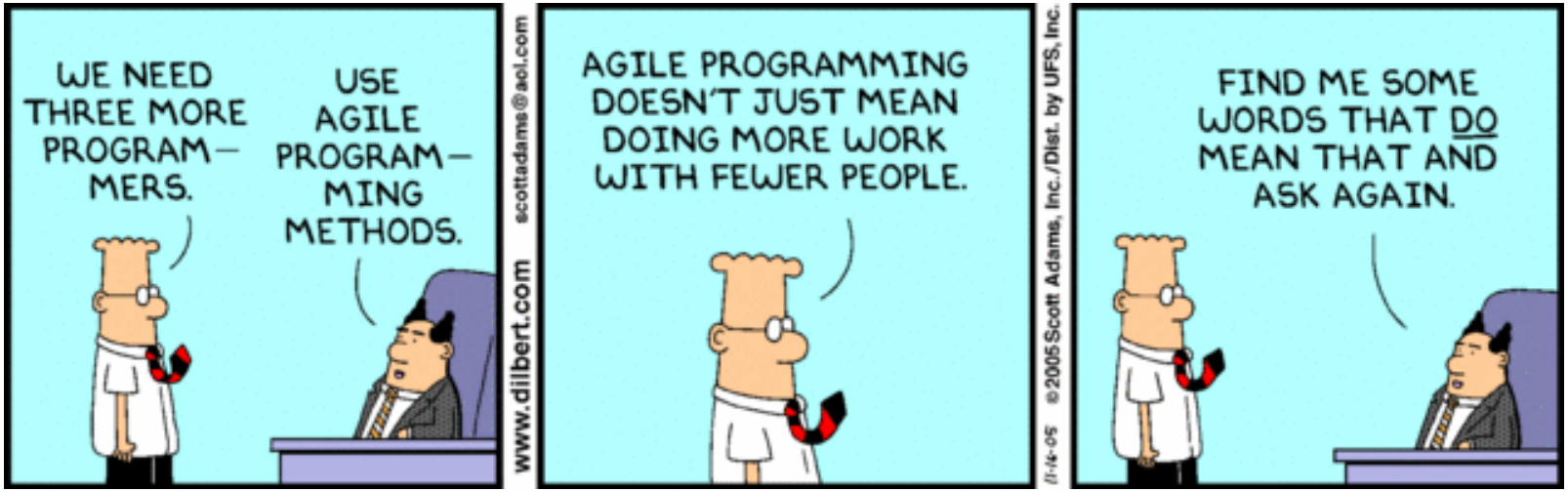
Some Reasons to go Agile:

- ✓ Easier, and less costly, to adapt to change
- ✓ Often better than Waterfall at managing risk
- ✓ Necessitates more involvement from stakeholders
- ✓ Developers feel more ownership. Empowered.
- ✓ Better chance of building something useful
- ✓ Quality-driven focus
- ✓ Greater transparency. Visibility.
- ✓ Quicker feedback from end-users
- ✓ It's actually kind of fun!
- ✓ ...?



Who Is Using Agile?

Scrum... Kanban...???



The Agile Drupalist

Flexible, Adaptable... like a Gymnast.

Okay all this theory is well and good, but how does this apply to Drupal development?

Drupal Strengths for Agile Development:

- ✓ Contrib modules help developers respond to changing requirements
- ✓ Drupal is a good tool for HTML prototyping
- ✓ Drupal works well with versioning (e.g. Git) and Continuous Integration (e.g. Jenkins), both of which are great for Agile development.
- ✓ ...?

The Agile Drupalist

Flexible, Adaptable... like a Gymnast.

An important point to note is that most often a (medium to large) Drupal website is going to be built by a web services agency.

A Typical Agency Project Life Cycle:

- ✓ Information Discovery
- ✓ Requirements Analysis & Information Architecture
- ✓ Design
- ✓ Development
- ✓ Content Creation (i.e. adding content to the site)
- ✓ Quality Assurance
- ✓ Deployment/Launch
- ✓ Maintenance

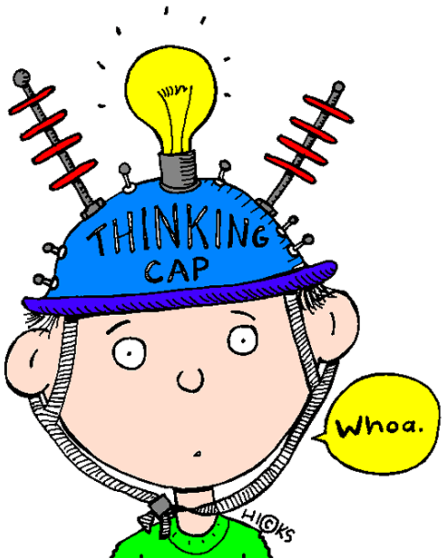
The Truth

You Can't Handle the Truth!

If you are an agency...

and you are developing in Drupal...

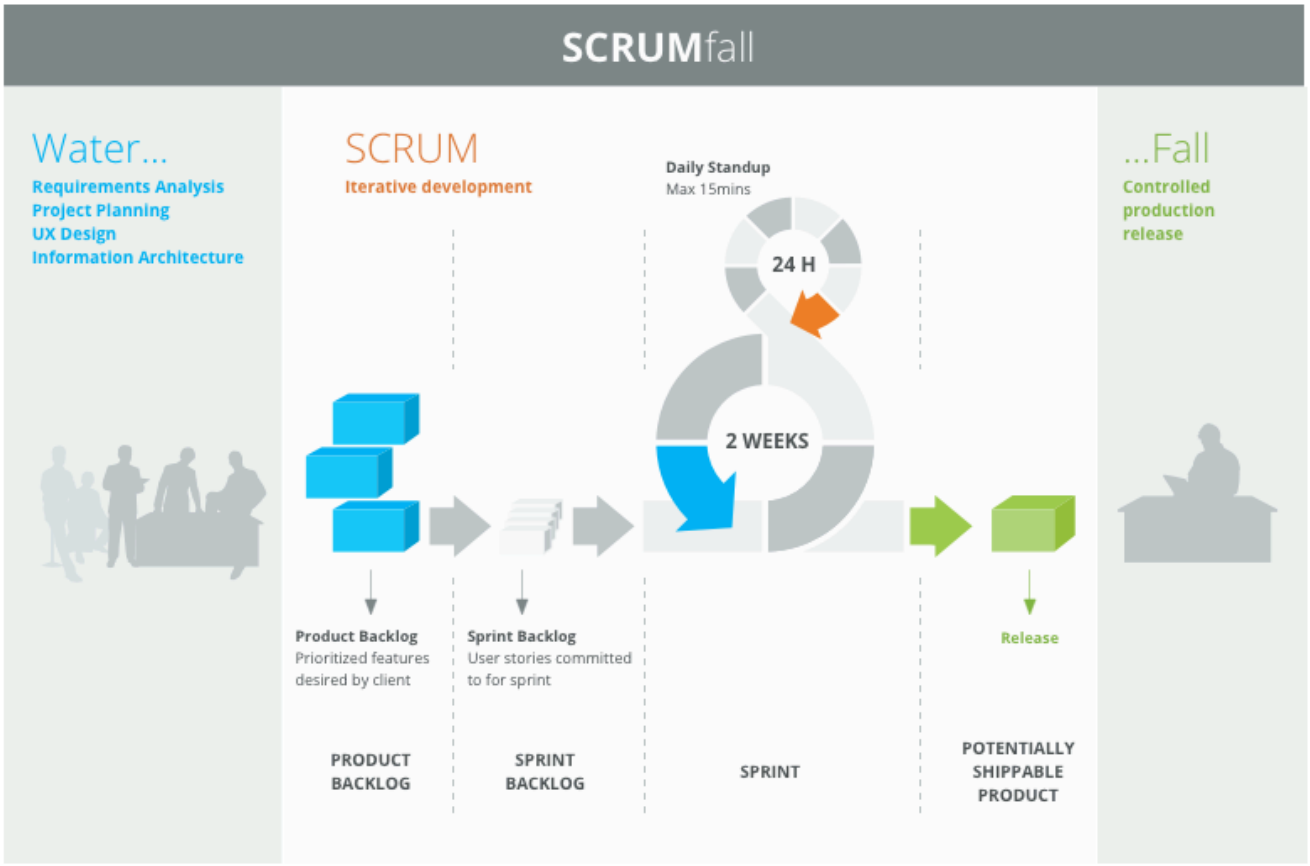
pure Agile is probably not the best option.



Scrumfall

No, it's not the title of the next James Bond movie...

Scrumfall is a "hybrid" of Waterfall and Scrum methodologies, often employed when there needs to be more certainty around the project scope.



Information Discovery

Phase #1 of 8

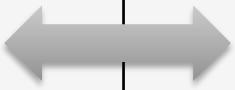
Waterfall Tasks	Agile Tasks
Documenting: • Project plan • Business requirements • Schedule • Risk analysis • Scope definition	Iterating: • Brainstorming • High-level requirements (i.e. vision, mission statement) • Elevator pitch • Product box

Requirements Analysis

Phase #2 of 8

Waterfall Tasks	Agile Tasks

Documenting:
 	
• Functional requirements
  • Content Types
  • Taxonomy
  • Sitemap
• Non-functional requirements
• User requirements
  • Roles & Permissions
	Iterating:
	
	• Low fidelity wireframes
	• Paper prototypes
	• Affinity diagramming
	• Use Cases
	• Process Flow Diagrams
	• UML Models



Design

Phase #3 of 8

Waterfall Tasks	Agile Tasks
Documenting: • Style and branding guide • Visual mockups	Iterating: • Higher fidelity wireframes including responsive layouts • Interactive designs

Development

Phase #4 of 8

Waterfall Tasks	Agile Tasks
Limited: • Building the initial architecture and infrastructure. • Maybe some of the groundwork for the high-level functional requirements. • Be careful of changing Content Types or Vocabularies!	• Breaking down epics into user stories. • Building out all the site features, modules, functionality in sprints by completing user stories. • Using the Features module to get things (i.e. entities) out of the database and into code so it can be versioned (e.g. Git).

Content Creation

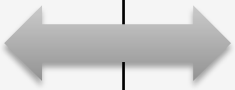
Phase #5 of 8

Waterfall Tasks	Agile Tasks
• Deciding at a high level what kind of content is going to be added to the site.	• Adding the actual content during sprints as features are developed.

Quality Assurance

Phase #6 of 8

Waterfall Tasks	Agile Tasks
• Matching finished site or site components with the functional requirements and overarching architecture. • Regression testing.	• Each user story should go through its own set of QA tests to satisfy the Definition of Done. • Product Owner reviews Acceptance Criteria and accepts or rejects user stories.



Deployment/Launch

Phase #7 of 8

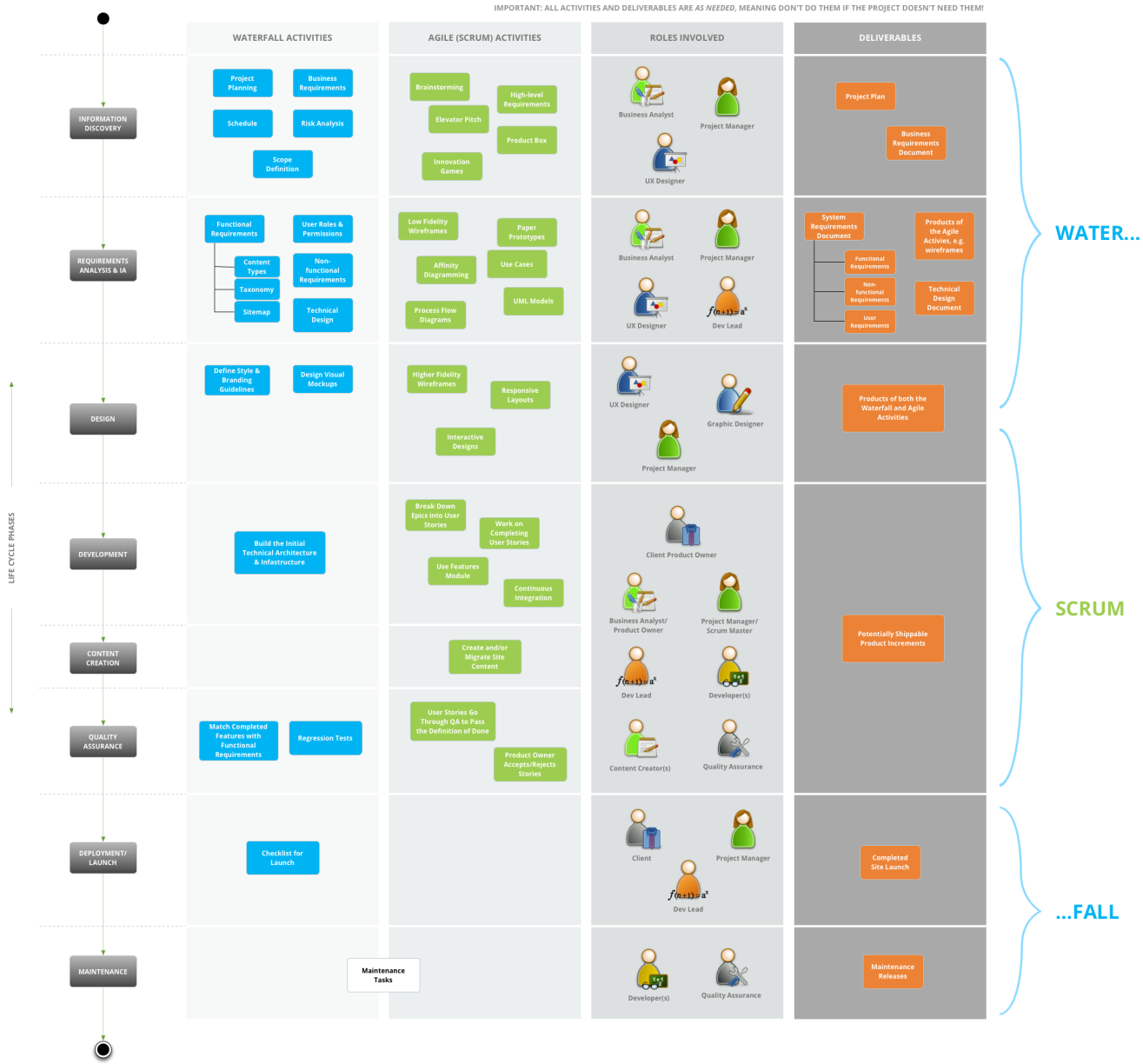
Waterfall Tasks	Agile Tasks
• Checklist for launch.	• Not as effective.

Maintenance

Phase #8 of 8

Waterfall Tasks	Agile Tasks
• No preference.	• No preference.

Putting It All Together

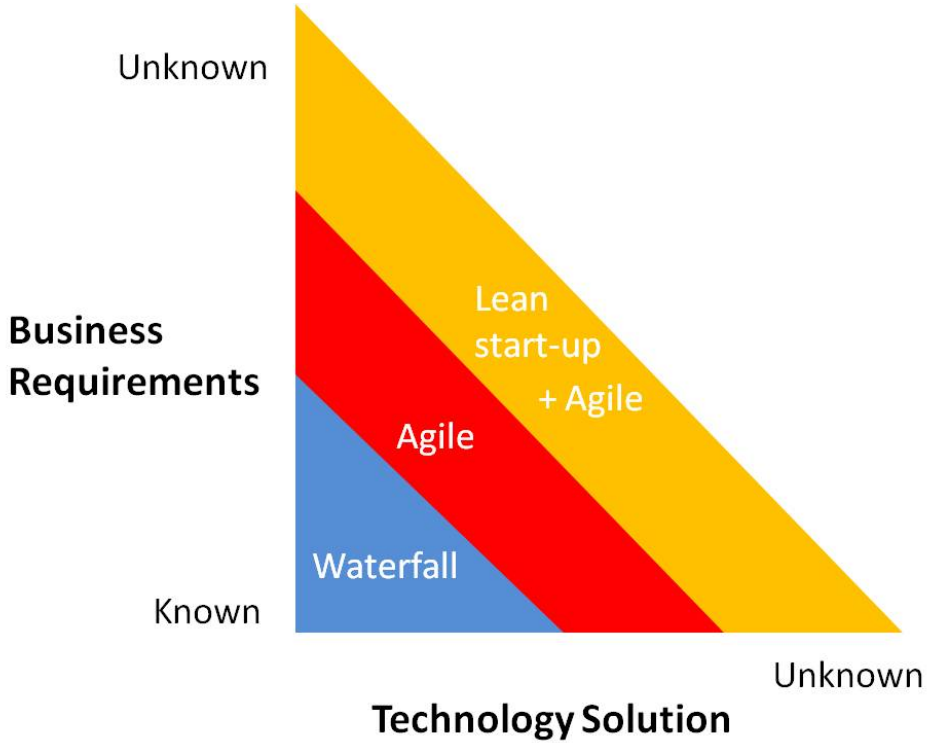


Things to Consider

To Scrum, or not to Scrum, that is the question

- Fixed budget? Fixed scope? You might want to be more Waterfall-ish.

- Is there uncertainty and the end goal isn't clear at the start? Is there a high likelihood of changing requirements? Agile can often get you a better result.



Real World Case Study

Putting yer money where yer mouth is

Redesigning Stanford University's **School of Engineering** website...

Case Study - Stanford

Waterfall Activities...

Business Requirements Doc	System Requirements Doc
Document Goals & Objectives: • Allow users to easily navigate between the main site and the different department sites in an integrated experience. • Provide easily accessible information to current students about graduation processes.	Functional Requirements: • FR-NCR-01: Display Latest News on Homepage. • FR-FSC-01: Faculty & Staff Directory Page. • FR-APR-01: Academic Programs Landing Page.

Case Study - Stanford

User Experience Design Activities...

Stanford | ENGINEERING

Persona: Meet *Suzy Smith* *(undergrad student)*

Category: Students

OVERVIEW

Full Name: Suzy Smith
Age: 19
Status: Undergrad (undeclared) Stanford sophomore
Technical Proficiency: An expert at using Google for searches, proficient with texting and mobile devices.
Social Media: Big on using Facebook to keep in touch with her friends and family back home. A well-known and active user on Instagram.

CHARACTER and PERSONALITY

About Suzy

- Sings in an *a cappella* group
- Assists with public service projects
- She is loving Stanford life
- From a small town, and is sometimes overwhelmed with the size of Stanford
- In-tune with technology
- She didn't come in knowing all of the engineering options

GOALS

What experience is she looking for?

- Wants to make the most of her time at Stanford
- Wants it to be an incredible experience
- Wants to solve the energy crisis
- Wants to change the world, and have an impact
- Wants to feel like SoE understands her needs

OBJECTIVES

What is Suzy trying to achieve?

- She wants to take an online class in engineering
- She wants to know when events are scheduled that are of interest to her
- Wants to know what dept is working on solving the world energy problems
- Wants to watch a lecture by Prof S. Bent from the Chemical Engineering department
- Wants to learn more about majoring in engineering and her study options

PAIN POINTS

What frustrates her?

- She finds it hard to find any information on the site about pursuing a degree that would allow her to work on solving the world's energy crisis
- She finds it hard to find faculty doing research on energy-related issues
- The site isn't mobile friendly
- She needs a more user-friendly site
- Finds it near impossible to find relevant event information
- How do you find person X's info?
- Doesn't know where to look to find a form to declare a major

ACTIONS

What is Suzy wanting to do on the website?

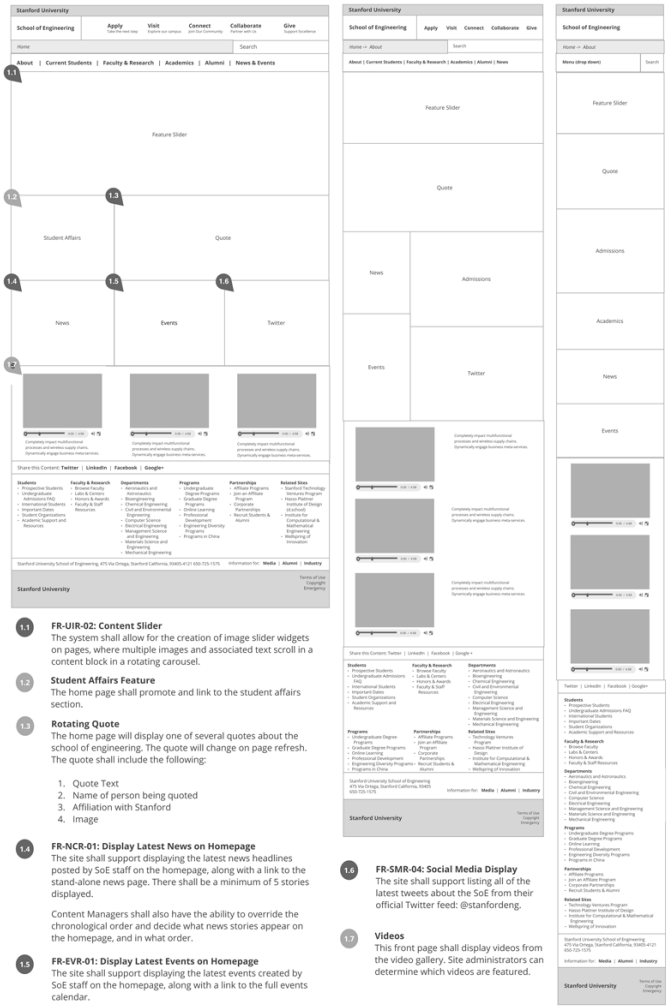
ACTION	TASKS
"I want to find out more information on academics."	- How do I transfer courses? - How do I switch my major? - What happened to a course (disappeared, cancelled)? - How do I petition program changes? - What are the steps to declaring a major? - Which department offers X study? - I want to file a grievance about my grade - I want to find out more about the China Program - I want to study overseas - How do I become a co-term (and what is it)? - What are the essential steps to graduating?
"I want to learn more about my education and my future."	- I want to learn about companies doing X - I want to learn how to be best prepared for industry - Do I need a Ph.D. to do X? - How do I take a class online (remote) - How do I get a TA appointment?
"I want to know what happens after I graduate."	- What kind of jobs do graduates get? - Who hires a Stanford student? - How do I find relevant job posts when I graduate? - Where do Stanford students get jobs? - If I was to go on and get a Ph.D., how would I apply for a post-doc at Stanford engineering? - What community volunteer opportunities are there for engineering students? - How do I get a summer internship? - How do I apply for work/study jobs?

LAST UPDATE
Thu Aug 22 2013

Case Study - Stanford

Wireframing Activities...

Stanford School of Engineering
Home Page



1.1 FR-UIR-02: Content Slider
The system shall allow for the creation of image slider widgets on pages, where multiple images and associated text scroll in a content block in a rotating carousel.

1.2 Student Affairs Feature
The home page shall promote and link to the student affairs section.

1.3 Rotating Quote
The home page will display one of several quotes about the school of engineering. The quote will change on page refresh. The quote shall include the following:

1. Quote Text
2. Name of person being quoted
3. Affiliation with Stanford
4. Image

1.4 FR-NCR-01: Display Latest News on Homepage
The site shall support displaying the latest news headlines posted by SoE staff on the homepage, along with a link to the stand-alone news page. There shall be a minimum of 5 stories displayed.

Content Managers shall also have the ability to override the chronological order and decide what news stories appear on the homepage, and in what order.

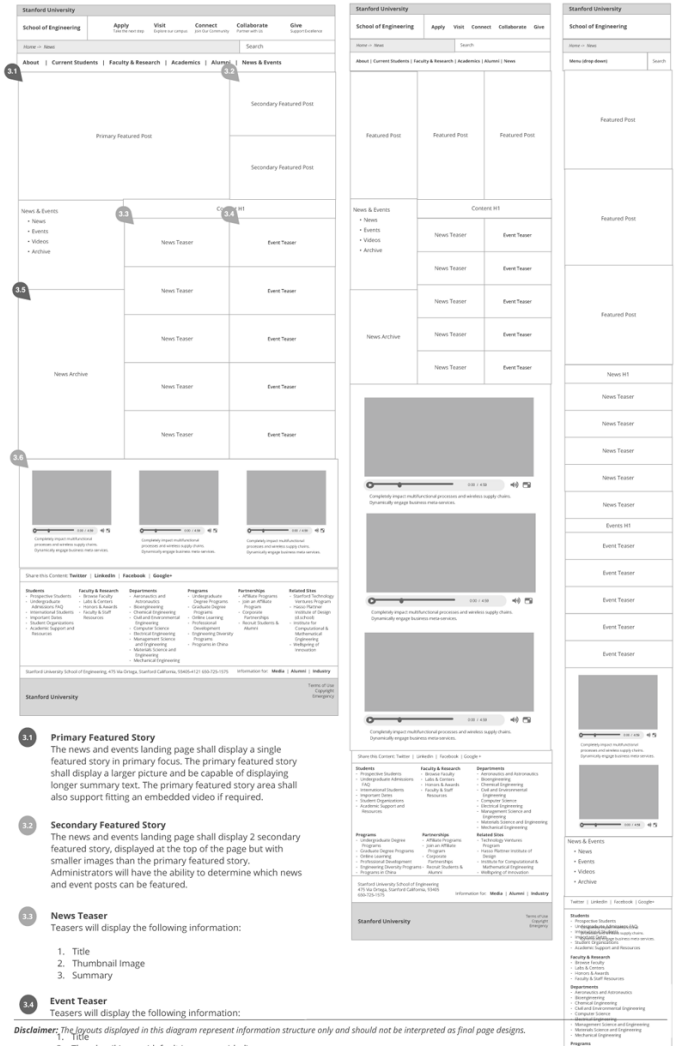
1.5 FR-EVR-01: Display Latest Events on Homepage
The site shall support displaying the latest events created by SoE staff on the homepage, along with a link to the full events calendar.

1.6 FR-SMR-04: Social Media Display
The site shall support listing all of the latest tweets about the SoE from their official Twitter feed: @stanfordeng.

1.7 Videos
This front page shall display videos from the video gallery. Site administrators can determine which videos are featured.

Disclaimer: *The layouts displayed in this diagram represent information structure only and should not be interpreted as final page designs.*

Stanford School of Engineering
News & Events Landing Page



3.1 Primary Featured Story
The news and events landing page shall display a single featured story in primary focus. The primary featured story shall display a larger picture and be capable of displaying longer summary text. The primary featured story area shall also support fitting an embedded video if required.

3.2 Secondary Featured Story
The news and events landing page shall display 2 secondary featured story, displayed at the top of the page but with smaller images than the primary featured story.
Administrators will have the ability to determine which news and event posts can be featured.

3.3 News Teaser
Teasers will display the following information:

1. Title
2. Thumbnail Image
3. Summary

3.4 Event Teaser
Teasers will display the following information:
1. Title

Disclaimer: *The layouts displayed in this diagram represent information structure only and should not be interpreted as final page designs.*

Case Study - Stanford

Into Development... Going Agile...

Stanford Team	ImageX Team
• Key Stakeholders • Product Owner	• Internal Product Owner • Scrum Master • Development Team

Functional Requirements

User Requirements

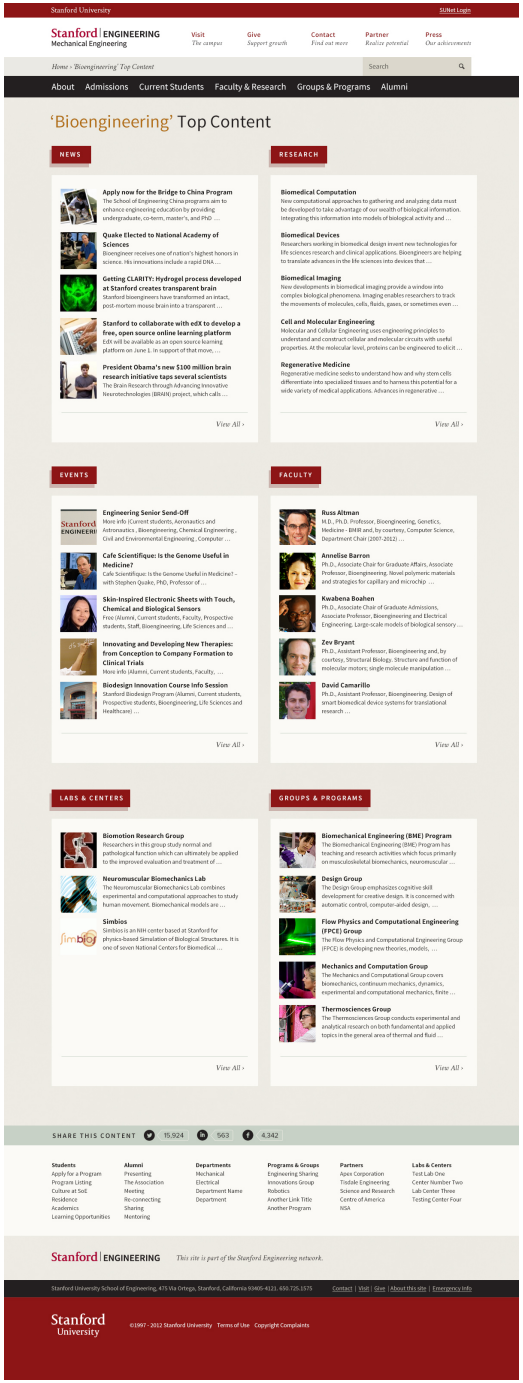
Epics & User Stories

Case Study - Stanford

Scrum Activities...

User Story:

"As a Stanford student I want to easily find information about bioengineering so that I can stay informed about the latest developments in this field"



BUILDING A SMARTER WEB

Thank You!

Any Questions?


@Adrian_D_Jones

adrian@imagexmedia.com