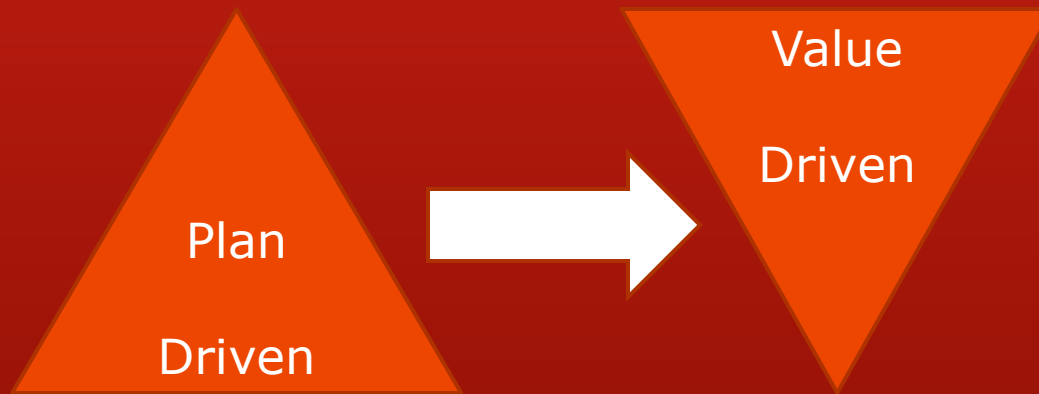


Migrating from Waterfall to Agile/Scrum



Greg Robleski
PMP, CRISC
October 2013



Who am I?



- Greg Robleski.
- MsTeri Communications (www.msteri-communications.com).
 - Deploying Open Source solutions to solve business issues.
 - Clients have included *EMC, Rainmaker, The Mayo Clinic, Kaiser-Permanente, The Central Arizona Project, American Express, Cisco, Dial Corporation, McKesson, Genomic Health, & Northrup Grumman.*
- Hold a PMP and a CRISC.
- 20+ years in technology and security management.



Why this presentation?

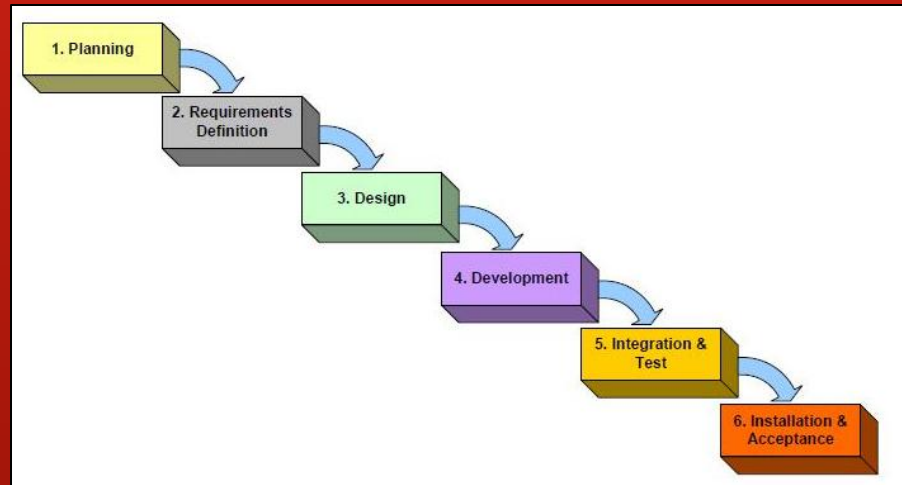


- Our number one challenge -- how to deliver better code on a faster timeline!
- Startup, emerging, and small companies have pioneered new approaches.
- Collectively they developed/embraced the Agile approaches.
- Now many Fortune 500 companies are trying to migrate away from their traditional Waterfall SDLC to Agile.
- This presentation is timely as it will outline the issues involved with migrating to an Agile Approach.

Today's Topics

- Recap of the Waterfall and Agile approaches.
- What & how to migrate.
- Transition enablers and roadblocks.
- Managing conflict/ resistance.
- Case Study.
- Questions.

What is the Waterfall SDLC?



Strengths:

- Traditional/Long-time use, ease of use, and easy to manage.

Weaknesses:

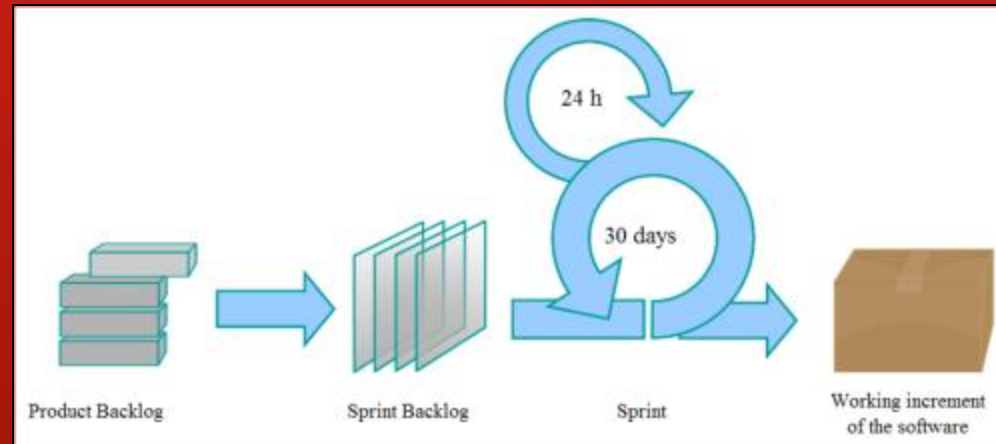
- Time to Market, too much thinking / not enough doing, and limited customer/end user engagement.

What is Agile?

Plan
Driven



Value
Driven



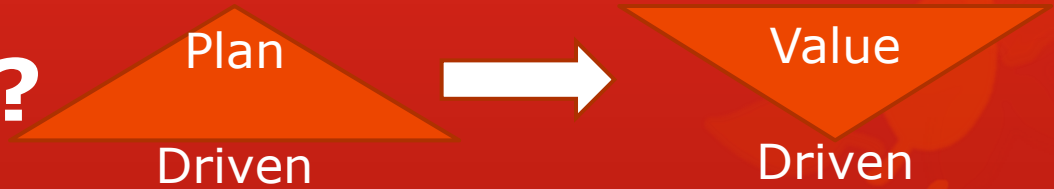
Strengths:

- Forces customer/user involvement, small/short development cycles, and ongoing learning/refactoring of code.

Weaknesses:

- Too many choices, runaway projects, and hard to implement.

What is Agile?



Defined by the Agile Manifesto - <http://agilemanifesto.org/>

- Four Core Principles that value:
 - Individuals and interactions over processes and tools.
 - Working software over comprehensive documentation.
 - Customer collaboration over contract negotiation.
 - Responding to change over following a plan.

Agile/Scrum



- Roles.
- Backlogs / Time-boxing:
 - Scope, not time is constrained.
- Sprints.
- Standups.
- Reflections/retrospectives.
- Deliverable, working code.



Deciding to Migrate – Why?



- Get stuff earlier....
 - ROI, Risk assessment / Cost waste reduction, Customer feedback, etc.
- Better 'quality' through earlier/integrated testing
- Customer satisfaction / Alignment.
- Emergent outcomes / Adaptive process.

Deciding to Migrate – Why?

- Earlier.....
 - ROI, Earlier Risk / Cost waste reduction
- Better quality' through earlier/integrated testing
- Customer satisfaction / Alignment.
- Emergent outcomes / Adaptive processes.

○ What you are currently doing isn't working!!!

Deciding to Migrate – What to examine?



○ Processes:

- Agile favors people over processes – is migrating your existing processes an oxymoron?

○ Culture:

- Is your culture supportive, risk taking, collaborative, curious, learning, etc.?
- Or is hierarchical, directive, protective, legacy-minded?

○ Style:

- Are you mechanistic or organic?
- Do you lead or do you manage software development?



Deciding to Migrate – How?



- “Big Bang” Approach:
 - Entire IT department and affected user departments - migrate all at once.
- Gradual / Phased:
 - Pick pilot projects to test drive Agile.
 - Pick and choose some elements of Agile/Scrum to adopt.
- Organic/ ‘bottom up’:
 - Let the working teams decide/embrace.

Migration Challenges - Pace



- Too high of expectations / panacea perception.
- Delivery of code will slow in the short-term.
- Adoption issues.
- Learning/knowledge issues.
- Resistance.
- Failure To Deliver **IMMEDIATELY!**

Migration Challenges – Enablers



- Vision and explanation.
- Cultural fit.
- Setting expectations.
- IT management/leaders.
- Business management/leaders
- Coaching/supportive and not directive approach

Migration Challenges - Roadblocks



- Cultural fit.
- People.
- FUD.
- Migration approach.
- Wrong skills / wrong time / wrong place.

Managing Conflict & Resistance



- **Need to change the discussion/ alter the paradigm:**
 - What one accepts as true (e.g., “The world is flat.”).
 - Prior Experiences/history.
 - Established knowledge vs. new/different information.
- **Some tactics:**
 - Address Personal Concerns first.
 - Link the Change to something.
 - Make the changes small at first, but especially concrete.

Case Study – Large Telecommunications Company



Background:

- Multi-billion dollar, multi-national telecommunications hardware, software, and services provider.
- Thousands of staff involved in product development, support and enhancements.
- IT staff scattered across the globe and roughly grouped into three development/testing centers.
- While business strategy is nimble, software development is/was pursued using a 'customized' Waterfall SDLC.
- Senior management fears falling behind because of its inability to deliver products and new features quickly to market.

Case Study – Large Telecommunications Company



The Issues:

- Senior management fears falling behind because of its inability to deliver products and new features quickly. **TIME TO MARKET.**
- Product quality (as measured by QA testing is/was falling. Identified large amounts of non-value added time (and staff). **COST/REWORK.**
- Moving to disparate, virtual teams. **LOWERED PRODUCTIVITY & COMMUNICATIONS ISSUES.**
- Too many 'runaway' projects. **WASTE.**

Case Study – Large Telecommunications Company



Approach:

- Selected Agile/Scrum as the way forward.
- Forced discontinuation of Waterfall as projects end. All new projects to be managed via Agile/Scrum.
- Hire new PMO agile/scrum trained staff and devolve the development organizations.
- Define product owners and business/technology structures and 'interfaces'.

Case Study – Large Telecommunications Company



Implementation:

- Rolled out to a selected division for testing and assessment.
- Rally selected as the communications/tracking/management tool.
- Developed a customized flavor of Agile, heavily Scrum-oriented.
- Formal training in tools and “customized” Agile.
- Emphasis on how to time boxing and assign backlog to releases/iterations.
- No user story training.
- No Fibonacci series/numbering. Rough T-Shirt sizing.

Case Study – Large Telecommunications Company



Results to date:

- Mixed bag.
- Some functionality delivered, but at a slower pace than previously.
- Business–Technology team conflicts/issues increased.
- Roles/responsibilities confusion.
- Not viewed as a success.
- Why?

Case Study – Large Telecommunications Company



Gaps:

- Estimation got worse, not better.
- iterations set too long – 6 weeks.
- Iteration was 4 weeks of coding and 2 weeks of testing – not integrated.
- Agile Commit meetings required 50+ slide deck – documentation reduction not achieved.
- Emphasis remained on specialists and not on grooming/developing generalists.
- Too many tools – MS Project, JIRA, and Rally.

Case Study – Large Telecommunications Company



Next?

- Re-rollout training.
- *Estimating, Estimating, Estimating....* Key to real success.
- Focus on driving everything into (and through) Rally.
- Establish product backlog – resolve user stories that cross product lines.
- Move to collocated teams and utilize a Scrum of Scrums concept

Thank You & Question/Answer

