Overhauling a Failed Project Using Out of the Box Scrum

2007 Fall Scrum Gathering
London, England

November 2007 | *version 1.2*

ajilus
the art of now

Objectives

Objective 1 – …how to become the new leader of an existing project and team without losing the historical team members and thought leaders

Objective 2 – …how to get the corporation to re-set goals and priorities and thereafter stimulate the company to equip teams for success by managing themselves

Objective 3 – …how to restructure the teams, responsibilities, leadership and reporting/statusing

Objective 4 – …how to deliver usable, tested software in a public forum and retrieve business/end-user validation

Objective 5 - …how to gain executive management respect and trust through transparency

How We'll Get There

- Biography Summary
- Company Profile
- Problem to be Solved
- Historical Timeline
- Turning Point
- Method of Overhaul
- Result Set
- Today's Behavior
- Questions

Biography Summary

As a managing partner and co-founder of a software consulting firm named Ajilus, Matthew D. Edwards focuses on helping customers deliver software simply and successfully with a low ceremony/high return team-based approach. From initial contract through development, delivery and support, Mr. Edwards concentrates his efforts on discovering and serving the customer's needs, building individuals, teams and leaders along the way, and working diligently to add discernible, tangible value by focusing on lower cost of acquisition and ownership solutions coupled with shorter return on investment windows.

Whether the problem to be solved is moving individuals, teams and companies into Agile spaces through training, coaching, teaching or consulting, solving problems of efficiency, effectiveness and quality, defining and leading on/off-shore ground-up software projects or overhauling those that are yet incomplete or failing, Mr. Edwards practices the belief that most software delivery challenges are more frequently socio-political and skill-based, and less often technical.

Through his career he has worked for start-ups to U.S. based Fortune 50 companies including Computer Sciences Corporation, Motorola and Nationwide in various team and leadership capacities during mergers and acquisitions, company launches, corporate conversions and new product launches. He currently holds certifications from the Quality Assurance Institute Worldwide in Software Testing (CSTE) and Software Quality Analysis (CSQA), (CMIIC) from the Institute of Configuration Management, is a Certified Scrum Practitioner (CSP), and holds a Master's degree in Diplomacy from Norwich University with a focus on international actors and interactivity frameworks.

Preface

- For the obvious reasons of proprietary confidentiality, all financial data and timelines are inexact in order to protect the privacy rights of the customer in question

Background

- A U.S. based software services company
- Is a software vendor for other corporations large and small
- Wants to revitalize business through a new ground-up product

- Communicated Goals:
  - Newer technology / platform
  - Improved capacity and scalability
  - Increased configuration flexibility
  - Decreased cost of customer acquisition
  - Decreased cost of customer ownership

Historical Timeline A

- Went outside for senior architect
- Architect came in-house to get staffing and found that people were simply too busy to support existing work and build a new application simultaneously
- Architect then went outside and assembled a team of individual consultants from different parts of the U.S. and other countries, all of them very competent individuals
- Estimated ~1 year timeline and US$1-1.5M

Historical Timeline B

- System should do everything existing system does, and more
- System should include market driven functionality that is anticipated, desired or otherwise existent in competitor products
- Decision was made to use new technology currently known to reside outside the skill-set of the existing employed staff
- System requirements were eventually defined/derived outside the customer's walls without inclusion of the customer
- There were no communicated or required checkpoints

Historical Timeline C

- The existing staff CIO/Development Manager began asking to be included in design and development direction and asking for status
- Eventually a consultant Project Manager was brought onto the customer site to manage the off-site project
- A Microsoft Project Schedule was created to show task hierarchy; status reports were provided directly to the President
- All members of the project were required to turn in written status reports which were summarized and later submitted

Historical Timeline D

- Status was reported in terms of a single page component architecture diagram using colors and percentages to tell the customer what was not started, in progress, and done
- Eventually statusing additionally included things such as number of source file checkins per week, number of hours worked, number of end-user screens mocked or attached to the application stack, or if defects had been worked on
- This worked for awhile longer and more staff were added to the project as requested in order to make more progress

Turning Point A

- For various reasons, eventually the staff CIO left the company, thereafter amplifying the desire of the customer to know ‘true’ status of the project but being left without anyone to peek in the window and understand what was seen
- Eventually, as internal staff leadership shifted, more requests were again made to ‘see’ what was being worked, delivered and communicated ‘done’
- As screens were not attached to the system, after nearly 1.5 years, the system did not yet possess significant features to demo to end-users other than discussing screen layouts, actors, activities, use cases, and reporting

Turning Point B

- At one point an external consultant was asked to look over the project and estimate what was ‘done’ and what was ‘left’
- Said estimates were based upon all forms of status available at the time including interviews, as well as, assessing those things asserted ‘done’ against the original use cases
- The estimate suggested, based upon how the project was planned and being executed in comparison to the definition of done, the system was on the road to spending US$3.5M before the communicated fundamental needs would be tangible

Turning Point C

- The customer requested options from the consultant with regards to how to complete the desired work in a compressed timeframe with minimized remaining cost

- The submitted proposal suggested
  - User stories and acceptance criteria to define ‘done’
  - Said stories to be prioritized/re-prioritized per sprint
  - 2 week delivery sprints of working, tested software
  - Sprint planning, sprint review, retrospectives
  - Modified leadership and status reporting mechanisms
  - 6 months or less to organize, execute and complete the ‘base’

Turning Point D

- What the project was doing well
  - Multiple daily builds (continuous integration framework)
  - Unit test and code paired and attached to the build routine
  - Team members were highly collaborative, highly motivated
  - Code quality was intelligent and thoughtful by the core team

- What the project was challenged to improve
  - Showing tangible, useful software that users could actually use
  - Making statuses meaningful so that non-technical business sponsors could understand what was done, what was left and where the money was going

- The project became more tense, but was left alone for multiple months 'as is' for fear of impacting progress - until the consultant Project Manager assigned to the project eventually left and the customer realized the US$3.4M reality

The Customer Request

- The customer again asked the third party consultant for a revised plan for taking over the project with a couple of particular goals
  - The project delivery behavior needed to come into alignment with how the existing product production environment behaved today (green-horn Scrum in evolution)
  - The internal production teams needed to come up to speed as quickly as possible on the new product and new technology so as to become productive with, own and support this product
  - The project was to be structured in such a way that the internal production staff could take over project leadership in less than three months
  - The customer wanted transparency in a manner understood by ‘all’

The Plan

- User stories/acceptance criteria were created, validated and prioritized for what was considered to be the fundamental base system that would generate demo opportunities and sales
- Teams were re-organized to mix consultants with internal resources - and internal resources were paired with consultants to facilitate the logistics, answer questions, and organize work, code, test and deliver
- Two sprint teams were created, a single prioritized work repository put in place (Xplanner), 2 week delivery sprints with sprint planning, sprint reviews and demos, user stories and acceptance criteria became the definition of 'done', and daily scrums replaced a litany of meetings and status reports

Time Budgeting Change

sp5 6/1-6/14									
People available for work during sprint	John	Rick	Mary	Ethel	Ron	Boris	Jen	Bob	Ted
Working days this sprint (pp)	10	9	9	10	5	9	10	10	10
Working hours this sprint (pp)	80	72	72	80	40	72	80	80	80
Support factor (% reserved for support)	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Drag factor (% unplanned lost time)	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Expected rework/retest hours (past+present)	8	8	8	8	8	8	8	8	8
Hands-on-keyboard time available for sprint	56	49.6	49.6	56	24	49.6	56	56	56
Total keyboard hours available this sprint		1128							

Challenges

- We needed to eliminate status conversation about component architecture, page mock progress, and measuring value in ‘hours’; and create conversation about user stories, acceptance criteria, relative sizing and burn-down charts using a ‘mix of’ for transition

- The project had picked up a co-located offshore development team alleged to not be producing, so new expectations, team composition and behaviors had to be defined to change the definition of ‘done’ and ‘done well’ and this was first managed at the executive level weekly, and then at the team level daily

- We needed the old team to want to stay, not leave prematurely or become apathetic, and be productive in terms of both delivering, and mentoring the internal staff for transition

The Results

- The third-party co-located consulting firm delivered at or above expectations for every sprint thereby enabling them to have their contract with the customer extended into the future
- The project delivered 103% of plan in the first two sprints, and 153% of plan by the end of four sprints
- Complete functional, tangible, tested software was demonstrated at the end of each sprint to anyone in the company with an interest
- Weekly executive meetings shifted to discussing user story priorities and progress, not dissatisfaction with not understanding progress
- Consulting members stayed, internal staff integrated and learned to lead and contribute, and internal staff took over leadership of the project in the expected timeframe

Example 'Status Snapshot'

Sprint Summary

	Sprint 1	Sprint 2	Sprint 3	Sprint 4
	Completed+	*Completed+*	*Completed+*	*Completed+*
Backlog *(h/#)* at Sprint Start	3300/119	2396/104	1751.6/61	1189.7/55
Accepted Into Sprint *(h/#)*	1067/37	1118.2/37	1185.8/36	1067/43
Current Actuals	--	--	--	--
Accepted Work Left	--	--	--	--
Capacity Left	--	--	--	--
Delivered in Sprint *(h/#)*	904/39	998/45	947.9/39	753.8/58
Backlog (h/#) at Sprint End	2701/104	1337/42	914.2/58	677.8/51
Stories Accepted	37	37	36	43
Stories Delivered	**39**	**45**	**39**	**58**
Things added to Backlog *(h/#)*	500/17	0/0	364.2/29	555.1/39

LKG:379

Calendar Remaining: 0%

Backlog Remaining: 20% (new deliverables)

Risk Areas: Hardware Availability

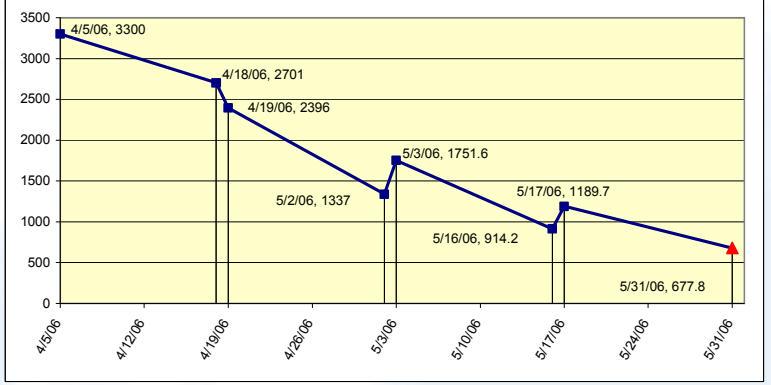
Planned (Hours/Stories) 3300/119

Actual (Hours/Stories) 3603.7/181

Sprint 1: 904h delivered / 250h added
Sprint 2: 998h delivered / 0h added

Effort Burndown History

Sprint 3: 947.9h delivered / 364.2h added
Sprint 4: 753.8h delivered / 555.1h added



Risks and Issues

Risk: A360 backlog for Sprints 5 and 6 exceeds available capacity for month of June and July. **Recommendation:** Accept it as a fact. Make sure everything is prioritized as well and current as possible. Make sure everything that is desired 'now' is decomposed into stories and acceptance criteria prior to Sprint 5 and 6 begin dates.

Issue: End state hardware stack not yet fully available and Geo5 installation not yet functionally usable. **Recommendation:** Accept as a fact and prepare teams to install end-state hardware installations as soon as available.

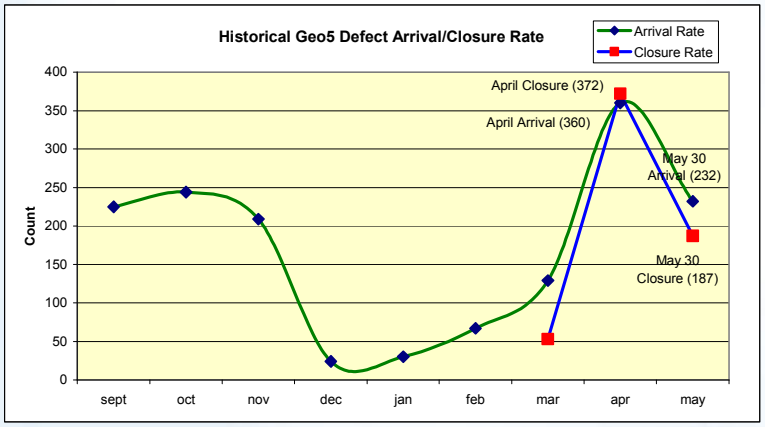
Functional Stability Summary	**Sprint 1**	**Sprint 2**	**Sprint 3**	**Sprint 4**
Backlog (Start:51)	*30*	*16*	*7*	*0*
In Progress	*2*	*7*	*8*	*21*
Under Review	*23*	*9*	*7*	*5*
Done	***0***	***23***	***33***	***29***

Batch 5: Rebuild Existing Postgres Scripts for MySQL+Linux Use COMPLETED

Snapshot	*Bld#*	*#Usrs*	*#Steps*	*Inc.*	*Duration*
A user can login	297	500	6	5s	59m
A user can search the catalog	297	500	11	5s	235m
A user can manage learning plan tmplts	306	300	22	5s	915m
A user can search their training	303	100	18	5s	39m
A user can register for a course	303	50	16	5s	157m
A user can manage user roles	297	500	38	5s	2009m
System: Instructor Report	321	50	10	5s	41m
System: Course Information Report	321	50	10	5s	48m
System: Registration Information Report	321	50	10	5s	40m
A user can launch online courses	339	500	20	5s	87m
A user can manage user groups	339	200	22	5s	835m
A user can create users	343	100	18	5s	435m
System: Student Information Report	348	20	10	5s	115m
A user can manage a course (create course)	356	200	23	5s	675m
10+ Boolean Report	356	50	10	5s	43m

Defect Arrival/Closure

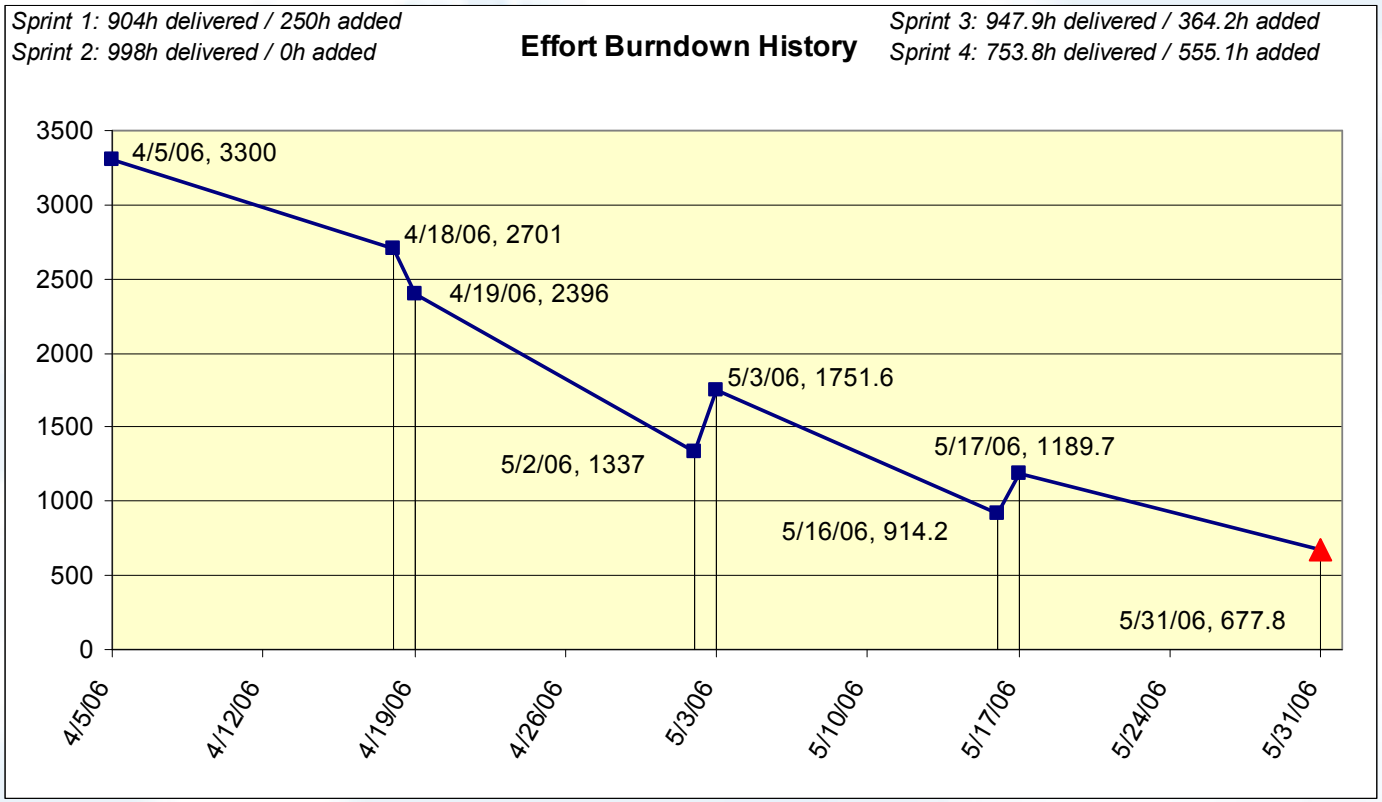
Today's Snapshot	*New,Assigned,ReOpened*					*Resolved,Verified,Closed*				
	S1	**S2**	**S3**	**S4**	Total	**S1**	**S2**	**S3**	**S4**	Total
ADD New Functionality	7	3	11	5	26	22	43	40	10	115
FIX Broken Functionality	19	59	26	14	118	321	537	241	68	1167
EXTEND Existing Functionality	--	6	6	4	16	3	15	50	19	87
Totals	26	68	43	23	160	346	595	331	97	1369



Example 'Sprint Summary'

Sprint Summary					
	Sprint 1	**Sprint 2**	**Sprint 3**	**Sprint 4**	**LKG:379**
	Completed+	*Completed+*	*Completed+*	*Completed+*	
Backlog *(h/#)* at Sprint Start	3300/119	2396/104	1751.6/61	1189.7/55	**Calendar Remaining:**
Accepted Into Sprint *(h/#)*	1067/37	1118.2/37	1185.8/36	1067/43	0%
Current Actuals	--	--	--	--	**Backlog Remaining:**
Accepted Work Left	--	--	--	--	20% (new deliverables)
Capacity Left	--	--	--	--	**Risk Areas:**
Delivered in Sprint *(h/#)*	904/39	998/45	947.9/39	753.8/58	Hardware Availability
Backlog (h/#) at Sprint End	2701/104	1337/42	914.2/58	677.8/51	**Planned (Hours/Stories)**
Stories Accepted	37	37	36	43	3300/119
Stories Delivered	**39**	**45**	**39**	**58**	**Actual (Hours/Stories)**
Things added to Backlog *(h/#)*	500/17	0/0	364.2/29	555.1/39	3603.7/181

Example 'Effort Burndown Snapshot'



Example 'Perf Testing Snapshot'

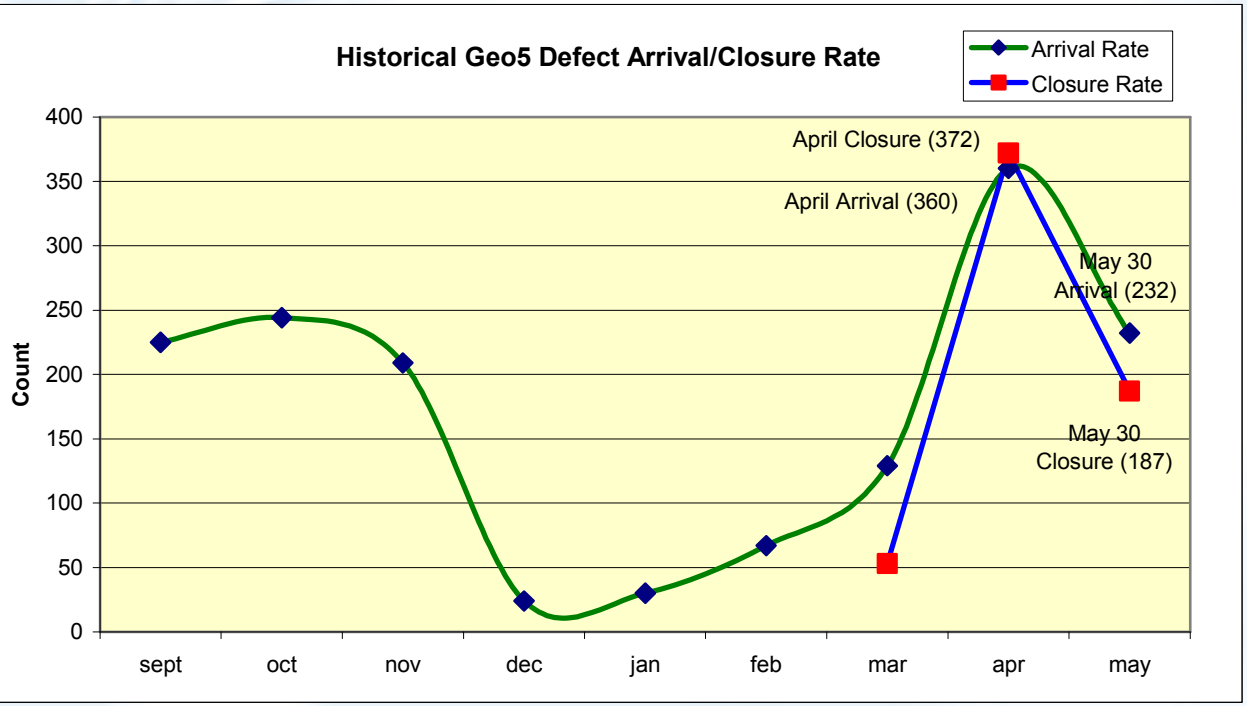
Batch 5: Rebuild Existing Postgres Scripts for MySQL+Linux Use COMPLETED

Snapshot	*Bld#*	*#Usrs*	*#Steps*	*Inc.*	*Duration*
A user can login	297	500	6	5s	59m
A user can search the catalog	297	500	11	5s	235m
A user can manage learning plan tmplts	306	300	22	5s	915m
A user can search their training	303	100	18	5s	39m
A user can register for a course	303	50	16	5s	157m
A user can manage user roles	297	500	38	5s	2009m
System: Instructor Report	321	50	10	5s	41m
System: Course Information Report	321	50	10	5s	48m
System: Registration Information Report	321	50	10	5s	40m
A user can launch online courses	339	500	20	5s	87m
A user can manage user groups	339	200	22	5s	835m
A user can create users	343	100	18	5s	435m
System: Student Information Report	348	20	10	5s	115m
A user can manage a course (create course)	356	200	23	5s	675m
10+ Boolean Report	356	50	10	5s	43m

Example 'Defect Arrival/Closure Snapshot'

Defect Arrival/Closure

Today's Snapshot	*New,Assigned,ReOpened*					*Resolved,Verified,Closed*				
	S1	**S2**	**S3**	**S4**	Total	**S1**	**S2**	**S3**	**S4**	Total
ADD New Functionality	7	3	11	5	26	22	43	40	10	115
FIX Broken Functionality	19	59	26	14	118	321	537	241	68	1167
EXTEND Existing Functionality	--	6	6	4	16	3	15	50	19	87
Totals	26	68	43	23	160	346	595	331	97	1369



Objectives Review 1

Objective 1 – …how to become the new leader of an existing project and team without losing the historical team members and thought leaders

- Achieved Through
  - Very many 1:1 phone calls, interactive messages, and email filled with “Thank you for your work” and “Well done” messages, and “We’re not done yet until the customer is transitioned, competent, and making money” conversations
  - Discussing the customer’s perceptions, comparing it to the external team’s perceptions, and affirming the team’s need to value and actively manage the customer’s perceptions at all times
  - Constantly asking the existing project team their perspectives on how to get ‘there’ and what we need to do for the customer to ‘come along’
  - Constant public acknowledgements, congratulations, thank you, and well done statements to the team and individuals every time it applies

Objectives Review 2

Objective 2 – …how to get the corporation to re-set goals and priorities and thereafter stimulate the company to equip teams for success by managing themselves

- Achieved Through
  - Getting to a single feature request list (in this case, a list of user stories) and getting the corporation to prioritize it every two weeks in preparation for the 'next sprint'
  - Getting the corporation to have the tough conversations of 'what is actually required to succeed' versus 'what do you just wish was there'
  - Showing the corporation a new way to measure 'done' through user stories and relative sizing, and then measuring progress through 'what was there, what is done, and what is left'
  - Especially on Sprint #1, Delivering what you signed up for; Repeat

Objectives Review 3

- Objective 3 – …how to restructure the teams, responsibilities, leadership and reporting/statusing

- Achieved Through
  - Over-communicating the need to serve the customer from the customers' value perspective as a pre-requisite to change
  - Practice pairing of Experts to Experts, Experts to Greenhorns
  - Keeping project and team leadership obviously and firmly grounded in the customers' geographical site ("Out of sight, Out of mind" sometimes breeds fear)
  - Co-location where possible, interactive messaging, email, phone calls, and daily Scrums otherwise
  - Build everything based upon the 'user story' as a definition of 'do' and 'done'

Objectives Review 4

- Objective 4 – …how to deliver usable, tested software in a public forum and retrieve business/end-user validation

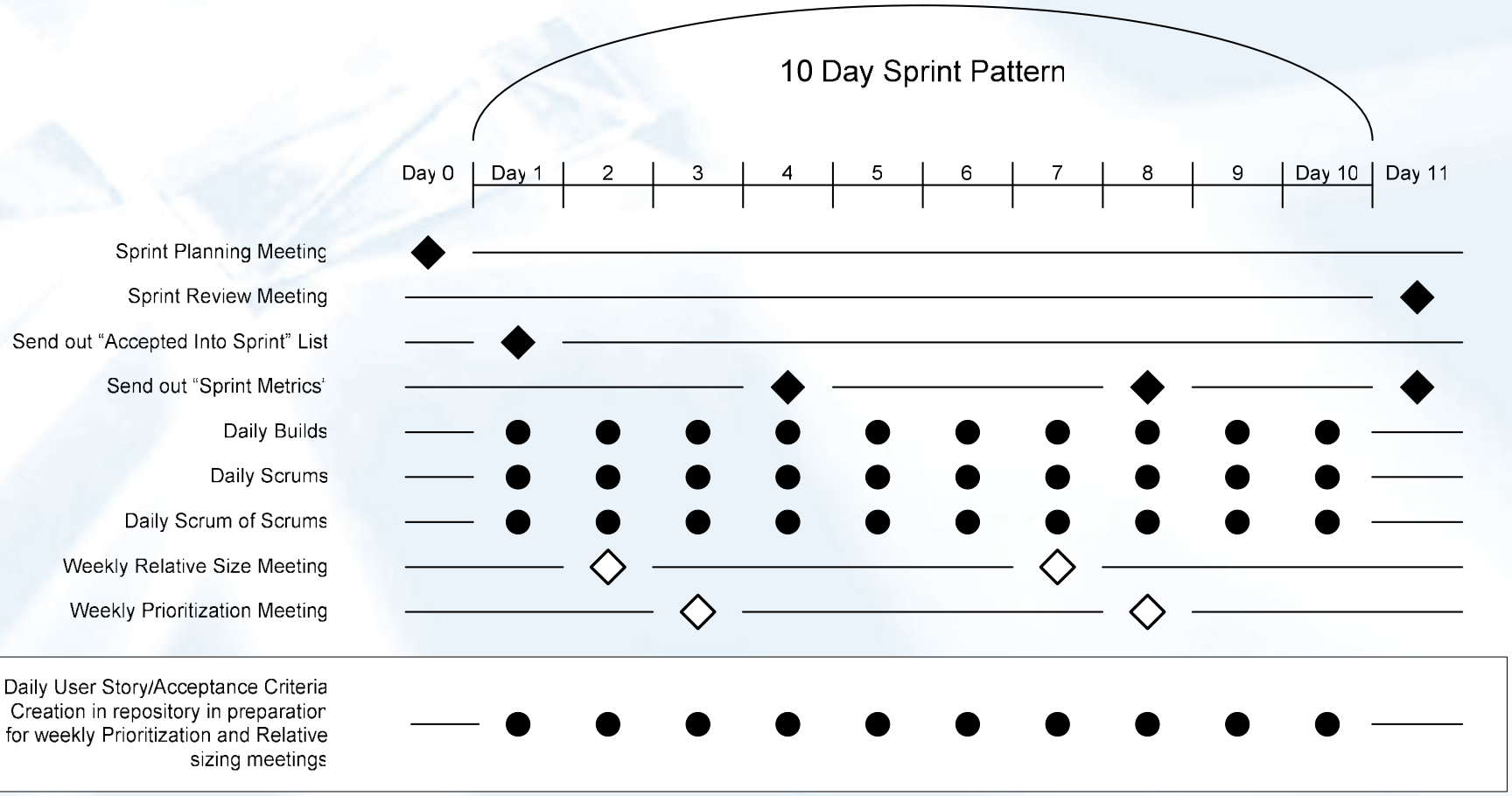
- Achieved Through
  - Having designated customer representative stakeholders in the user story definition process, sprint planning meeting as a chicken, and sprint review meeting as a ‘closed-loop validator’
  - Holding end of sprint demonstrations where the person who did the work, did the demo, and anyone with questions could ask, but only designated customer stakeholders could submit change observations/requests which may become work for ‘next’ sprint

Objectives Review 5

- Objective 5 - …how to gain executive management respect and trust through transparency

- Achieved Through
  - Near daily conversations regarding impressions, risks, issues, successes and failures (walk-bys, drive-bys, after hours calls, anytime email, etc.)
  - Making the project status something everyone can understand which is…
    - "As a user, I want to…"
    - Reported using "Burn-downs built on said user stories"
  - Minimizing non-useful meetings, statuses and general 'chaff' conversation; ensuring the primary meeting with executive management is transparent, quick, to the point, and over
  - Inviting questions and challenges any time

Example Sprint Pattern (after some Maturity)



Contact Data

Matthew D Edwards
Ajilus, The Art of Now

matthew@ajilus.com
www.ajilus.com