The Evolution of a Super Agile Software Development Capability

D R A F T E

C. C. Shelley

September 2011

Oxford Software Engineering
www.osel.co.uk
9 Spinners Court, 53 West End, Witney, Oxfordshire, England, OX28 1NH
shelley@osel.netkonect.co.uk

ABSTRACT

This paper describes how a novel, possibly unique, software development capability was evolved from a baseline of diverse agile software development practices. ***bwin****, an Austrian software organization with demanding requirements of its IT department has, by using well established technology transfer practices, acquired an extremely predictable and scalable 'Scrumbled CMMI' software development capability. Data extracted from the tools and infrastructure supporting this new capability show how it developed over a 30 month period from 2007 to 2009.*

1 Introduction

Agile software development practices[1] emerged during the 1990's as a reaction to the inappropriate application of software project management[2] They are now familiar, at least by reputation, and some remarkable successes have been reported.

The continued interest in agile software development is to be welcomed, although the uncritical enthusiasm for ill defined 'agile' (no noun) makes a disinterested evaluation difficult, the re-encroachment of project management as 'agile project management' is a concern, and, as with any new ideas, the take up of agile practices is difficult and can be 'patchy'. (see www.osel.co.uk/whatsnew.htm#TomonGerry). The uncritical enthusiasm for agile practices obscures their very real limitations and may in time lead to a backlash and rejection of this valuable, new(ish) approach to software development. However, some organizations have taken a mature and pragmatic view of the benefits and limitations of software development agility and have carefully adopted and adapted those elements that fit their commercial needs. The case study reported here is one such case. Starting from a set of Scrum like practices ***bwin*** directed the development of these practices into something unique. This is described below.

2 Background

bwin is a European company providing online entertainment, sporting and betting services. It can have upwards of 250,000 concurrent users. It is multi-transactional and international, providing services in more than 20 languages, and deals with more than one million transactions per day.

It requires secure implementation for its financial transactions and must respond rapidly to changing commercial opportunities and to changing national legal requirements.

[1]Usually thought of as Scrum and to a lesser extent eXtreme Programming, generally conforming to the ideas within the agile manifesto

[2] Not, as often suggested 'waterfall' type software development lifecycles'. Waterfall and other Big Design Up Front (BDUF) models are often suggested because they tend to be the default lifecycle for software projects, whether or not a project oriented approach or a waterfall lifecycle is appropriate.. Much, perhaps most, development work is ill suited to this project oriented approach but until the advent of agile software development there were few widely known, acceptable or attractive alternatives.

bwin has two development sites. One in Vienna and the other in Stockholm. Technical staff at both these sites are organized into 'delivery teams' that develop and support the systems. These delivery teams are supported by the architecture team that attempts to control shape of bwin's systems, and by a team of test specialists that are embedding within the delivery teams.

Originally these teams used traditional development practices until agile practices (essentially Scrum) were introduced and encouraged by senior management. The introduction of these practices were a natural evolution of development practice meeting their requirement for a responsive and flexible approach to supporting their systems; fixing defects, adding enhancements and introducing changes frequently, dependably, and regularly.

The conventional Scrum approach was modified to better meet business and depended delivery teams needs. These changes included:

- allowing changes to be introduced mid sprint to provide higher levels of responsiveness,
- varying the length of sprints to ensure required stories are included at the end of the sprint,
- scrums producing and communicating widely daily re-estimates of the sprint,
- external monitoring to provide high levels of predictability.

These 'super agile' practices have been used by bwin to provide the flexibility and responsiveness required by the business. They were developed and evolved to fit delivery teams needs. Each of the delivery teams developed their own process to meet their particular needs.

However serious problems remained. Delivery teams were (and still are) dependent on each other. When a team failed to deliver what was expected when it was expected this impacted on teams 'upstream' that were expecting those deliverables. Not only this, because delivery teams had evolved their own ways of working, suited to their particular needs, there were not well understood beyond the team, so communication of development or technical issues was limited. Team activity and issues were 'opaque' to other teams and to the business. Cascading delays caused inter team and team/business friction. While teams' ways of working were improving it was by slow, undirected evolution.

3 Transforming the delivery process

bwin could no longer afford these delivery delays or surprizes. A radical improvement in the capability of the bwin's IT capability was required. The business required increased responsiveness to rapidly changing business need, both tactically and strategically, unhindered by unexpected and cascading delays.

It was recognized that by improving the capability of each of individual delivery teams, this would, in turn, would enable an improved overall collaborative capability. A decision was taken to accelerate and direct the evolution of bwin's delivery processes.

This new capability would be founded on high levels of delivery predictability. All of bwin's delivery teams would become highly predictable. This prime objective, to improve all teams' ability to consistently deliver both *what* was expected *when* it was expected is unlike traditional scrum or agile development with its focus on *when*, and often quite poor performance delivering *what*.).

It was understood that it was critical that these developments in delivery capability were not to be a trade off . Improved predictability must not impact systems quality or increase costs.

It was anticipated that this foundation of a uniformly high level of delivery predictability would enable a novel strategic development capability: to make major system changes, and implementation of new systems very rapidly with teams able to work collaboratively. Predictability for individual teams enabling teams teams to synchronize work and deliver concurrently, and dependably to a common 'rhythm'.

This synchronization and concurrency, with teams working with a common process provides another capability – scalability. Additional teams can be added to major development work to speed it up. These teams may be extant, and simply re-assigned from other work, or may be newly formed.

(Note: The assignment of dependable delivery as the top priority improvement goal is unusual. Many attempts to improve development capability have unclear or simplistic ideas of improved 'productivity' or 'quality'. bwin had a very clear idea of what was wanted - and for a very specific reason.)

The way to deliver this high level of delivery predictability was to be a shared or common delivery process used within all delivery teams. This common process, used consistently by all delivery teams and understood by the business, with well understood interfaces, would enable shared understanding of every team's delivery status, its dependencies and risks at any time. It would provide early visibility of emerging issues and hence low friction in inter team and team/business working.

This new, common delivery process had to be designed and refined to have special characteristics. Particular attention was paid to estimation at the beginning of a sprint. Unlike most agile practices where the delivery schedule is tightly managed or time-boxed but the value delivered, ordered by

priority, is variable, bwin's process needed very good control of what was delivered as well as when Considerable care in estimation before committing to delivery was an important part of this. Similar care was put into developing an effect process for recognizing, identifying and managing dependencies. An essential characteristic, unusual in agile practice, which deprecates 'micro management', was a requirement for 'process transparency'. It was essential that at any time during a sprint the status of the sprint - including risks, emerging issues and other matters that may impact delivery - would be visible to others outside the team: dependent teams and other business stakeholders (all of which were carefully identified).

The work to direct the evolution of diverse delivery teams' diverse practices towards this common process was not easy. There were several false starts, and both agile coaches and process experts occasionally struggled recognize the needs of the business above and beyond process engineering's conventional wisdom or agile dogma.

Taking cues from the ideas within CMMI, in particular the practices and characteristics of a maturity level 3 type organization delivery team members worked together to evolve a widely accepted common process. This development of a 'Scrumbled CMMI' is essentially a complement to the work described by Jeff Sutherland in *'Scrum and CMMI Level 5: The Magic Potion for Code Warriors'*. Sutherland reports the effects of introducing Scrum into a high maturity defence contractor. bwin have introduced CMMI type process thinking into an agile environment with similarly remarkable results. This suggests that judicious 'mix and match' of reference models can deliver impressive outcomes so long as the objectives are well understood.

Important, critical even, factors in the successful evolution of this acceptable and effective common delivery process work were:

- Collective, consensual working: Delivery team staff themselves worked to develop the common process – developers, testers and managers.

- They identified common elements in their current, diverse ways of working, selecting and refining elements to contribute to the common process as required and developing new elements as required. 'Better' ways of working were not imposed from outside by management or process experts.

- They *used* generic 'reference models' (CMMI and Scrum) to to develop the bwin 'operational model' (the common delivery process) rather than simply mimicking or complying with the reference models.

With the emergence of a common delivery process, tuned to provide high predictability (by giving close attention to estimation and dependency management) this process could then be supported by a set of carefully selected and adapted state of the art software tools. A configuration management system, based on Subversion was fundamental; enabling continuous integration and daily builds. A number of automated software testing and quality tools were integrated. A management system, built around Version One, provided a common environment to organize and manage business and sprint backlogs, support technical work and provide a reporting infrastructure for the delivery teams, managers and the business. This environment routinely captures data from all teams providing a consistent, near real time picture of team status and software quality at all times. As well as providing support for the common process this tool set also reinforces or 'institutionalizes' the bwin delivery process too.

4 Results

This section contains part of a report submitted by bwin to the IEEE and SEI as part of a submission for a software process award. No award was won, but the report was resubmitted to the later used in bwin's submission to the European Software Institute as a submission for the ESI's 'Software Excellence Award'. bwin came third in its class.

The annex is reproduced as submitted. Where necessary informative comments have been inserted in italics:

Background

Data Data presented here is for delivery teams using the DM Process. It is derived from DM release spreadsheets for 2007 to 2009 and defect data from AC Pro from mid 2007 until August 2009. Earlier, unclassified defect data has not been used, and no defect data from VersionOne (the recent successor to AC Pro) is included here.

Analyses presented here have been kept very simple and attempts have been made to allow the data to speak for themselves. Most are graphical, arithmetic is kept to a minimum and few assumptions need be made. No filtering or selection has been made, outliers have been left in place, even when they spoil the pattern or trend and could be legitimately removed. Readers may wish to investigate these outliers.

These data reflect the performance of the DM process as used by delivery teams. No selection or stratification by teams or products has been performed.

During 2007 six teams were using the standard, documented DM process, and ten were not, during 2008 ten teams were using the process and six were not. In 2009 all sixteen teams were using the process. To reflect this null data points have been inserted in the data: five for every three points in 2007, 3 for every five points 2008 and none in 2009. Thus the sparseness of the data reflects the limited use of the process in the early days.

The process improvement work (undertaken by the "R2D" project) to standardize delivery team work on a common process was undertaken to align delivery team's work to the ***bwin*** business drivers, and achieve clear, measurable performance objectives.

bwin requires its teams to be dependable 'delivery units' that will delivery changes and enhancements with a high level of predictability. These dependable 'delivery units', working together, provide a software production 'carrier' that enable ***bwin*** to respond to rapid and unexpected changes in a global market.

In priority order the business drivers the DM process must satisfy are:

Predictability

It is critical that the DM process delivers changes to the ***bwin*** systems predictably. That is: changes are delivered as expected and when expected. ***bwin*** must be confident that changes that delivery teams have committed to deliver will be delivered, or that failure to deliver will be communicated at the earliest opportunity.

Bwin's process improvement work aimed to *increase* DM process predictability.

Responsiveness

Bwin operated in a rapidly changing commercial and legal environment. From time to time the ability to commit to and deliver unpredictable and urgently needed changes to the ***bwin*** systems is highly desirable and, on occasion, essential.

The ***bwin*** process improvement work aims to *sustain* DM process responsiveness.

Frequency and Regularity

Bwin systems require many and frequent changes. The DM process must deliver changes to systems frequently and regularly, establishing dependable delivery 'rhythm', familiar to both delivery teams and the business.

The ***bwin*** process improvement work aims to *improve* current DM process frequency and regularity.

These will improve if schedule (sprint) predictability improves.

Systems Quality

The ***bwin*** systems operate in a competitive environment where certain qualities are important. In particular:

The ***bwin*** process improvement work aims to *sustain* the current, manageable systems quality.

Predictability Performance Indicators and Targets

The essence of these performance units is predictability. ***Bwin*** has developed a set of measures of predictability and delivery team and organizational targets:

1. Schedule predictability is measured by the number of days after planned date of delivery
2. A team's success factor for schedule predictability is for a delivery to be within seven days of the planned date. If successful a flag is set to '1', if more than seven days then the flag is set to '0'.

3. The ***bwin*** performance target for schedule predictability is for 90% of deliveries to be within the seven days margin, in other words, 90% of flags set to '1'.
4. Predictability also involves delivering what was expected - scope. Schedule adherence but with only part of what was expected is not acceptable. Teams commit to deliver a number of stories comprising a number of storypoints[1].
5. A team's success factor for scope predictability is for 90% of committed storypoints to be delivered. If more than 90% of storypoints are delivered then a 'scope flag' is set to '1', if less than 90% then the flag is set to '0'.
6. The ***bwin*** performance target for scope predictability is for 90% of deliveries to be within the 90% scope margin, in other words, 90% of flags set to '1'.

(Note 1: Storypoints are ill defined. They are used as part of agile estimation and are considered more as a locally calibrated, precursor or proxy to an effort estimate. However, like previous measures of this type the boundary between effort or resources consumed, and software or valued delivered is murky. Within bwin storypoints were used both as an indication of valued delivered to the business, i.e parts of stories, as well as the melded measure of effort, skill, complexity, etc used to deliver them.)

Schedule Predictability

A KPI has been designed to monitor schedule predictability. Deliveries within seven days of planned delivery date are successful. A target for 90% deliveries to be successful has been set – and achieved.

Note 2: The graphics presented below have been produced quickly and simply as part of an exploratory analysis process. In general graphics are no more than 'three clicks' from the data. They were and are not intended as 'presentation' graphics, but simply required to illustrate, without fuss, any trends or patterns.)

(Note 3: Trend lines have been added later to this and other graphs in an attempt to high light 'trends'. These are often misleading or erroneous , especially were data are categorical in nature. They should be ignored.

Predictability KPI:

The graph below is a sequence of all deliveries from 2007 to 2009. Successful delivery is indicated by a '1,' unsuccessful, by a '0'.

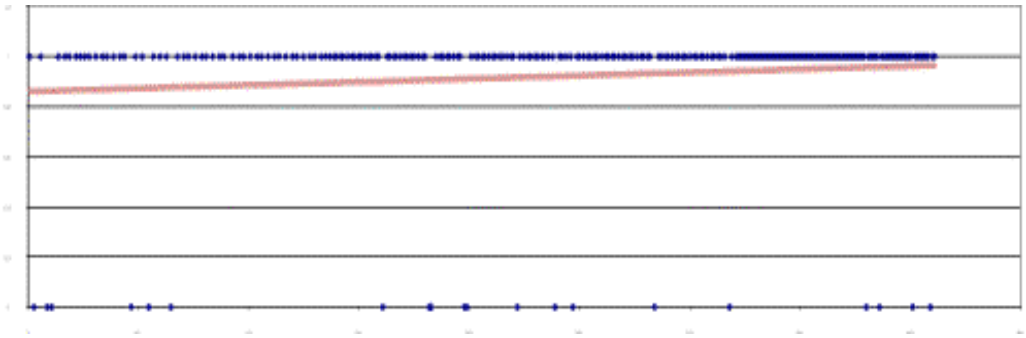

Figure 4.1 – Schedule predictability KPI indicators for deliveries from April 2007 to August 2009.

Note how the density of successful deliveries (on the '1' line) increases over time, but the density of unsuccessful deliveries (on the '0' line) remains more or less constant. The ratio of successful to unsuccessful deliveries has increased.

Predictability as measured by days delay

The next graph shows the actual number of days deviation from the planned delivery date for the same set of deliveries:

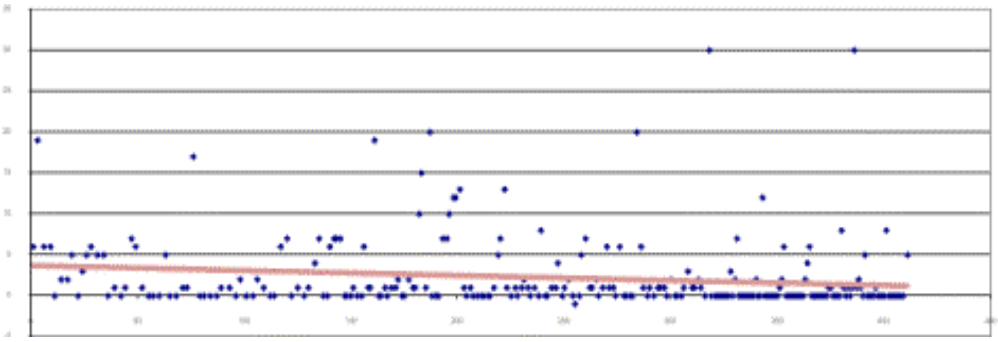

Figure 4.2 – Days delay for deliveries from April 2007 to August 2009

Again the improvement in predictability is clear – note the increasing numbers of data points on the zero days line during 2009.

(Note: Predictability is limited to a delivery being 'on time' and unpredictability as 'late'. Only one instance of early delivery is recorded. This suggests that, in practice the real issue is not simple predictability, which could be expected to show a more balanced distribution of data about the 'on time' date. (Why are all deviations from the planned date late, not early? It may also suggest unmeasured 'slack' in the system when DM teams do, in fact finish ahead of schedule but do not 'officially' finish until the planned date? To be investigated?))

The Improvement in DM process predictability can be summarized in the following way:

In the first four months for which we have data 6 of 20 releases were within one day of planned (no more than a day late). For the last four months for which data are available 36 of 42 releases were within one day of planned. This gives a ration of about 13:36. That is about a **three times improvement in schedule predictability**.

Looking at the number of days delay:

If the number of days delay for releases in the first four months are considered then the average delay per release is 4.25 (85 days delay/20 releases). For the last four months the average is 1.45 days per release (61/42), and if the exceptional delay of 30 days, for SP 09-06 FRB 3.25, is removed this reduced to 0.74 days per release. The ratio between the first four and last four months is a **2.93 times improvement in schedule predictability**, or, excluding a SP 09-06 FRB 3.25, a **5.74 increase in schedule predictability**

Scope Predictability

Predictability (as required by ***bwin***) is not just delivering on time. It is delivering *what was expected* on time. That is, if a team does deliver on time, but does not deliver all the expected stories – especially those that other teams will depend on [2] - then its value is reduced.

Thus the proportion of delivered story points (compared to those committed) is a critical dimension of predictability.

The performance indicator for scope predictability is the percentage of committed storypoints actually delivered. More than 90% delivered indicates success. Figure 5.1 shows success data for 2008 and 2009. Success is indicated by 1, failure by 0.

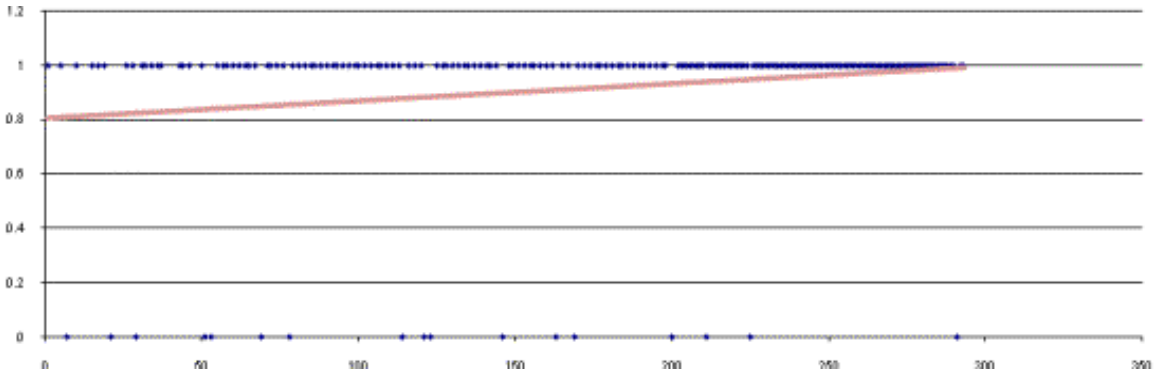

Figure 5.1 – Scope KPI indicators for deliveries from January 2008 to August 2009.

It can be seen that as the frequency of unsuccessful deliveries declines wit time the proportion of successful (within scope deliveries) has increased during 2008 – 2009.

The Improvement in scope predictability can be summarized in the following way:

In the first four months for which we have data (January – April 2008) 14 of 20 releases achieved the 90% scope target (with 6 failing and no data for 7 others recorded.) For the last four months for which data is available (May to August 2009) 39 of 40 releases succeeded. This gives a ration of about 12:1. That is about a **twelve times improvement in scope predictability**

(Note 4: The summary figure of improvements to predictability was requested by bwin's management to provide concise and compelling 'headline figures'. These, and similar figures reported below, while accurate should be treated with care: Initially the first and last three months figures were calculated and compared. These indicated an eight fold improvement. This was considered excessive and lacking credibility (despite being correct). Therefore the first and last four months were calculated and compared. This produced the exciting and credible five fold improvement. If the first and last six months figures had been used a more modest improvement could have been reported – all from the same data. This illustrates the need for considerable care in interpreting 'improvement' data and the value of graphics in providing a broader, but perhaps less compelling picture of improvements.)

Storypoints

An examination of the numbers of storypoints committed and delivered shows scope predictability in detail.

(We know that story points are not the best measure of delivered value, stories are, but storypoints will be correlated to stories)

The graph below shows the ratio of committed to and delivered storypoints.

The banding is due to the inclusion of null data points as described above. The ratio is shown better if these are removed:

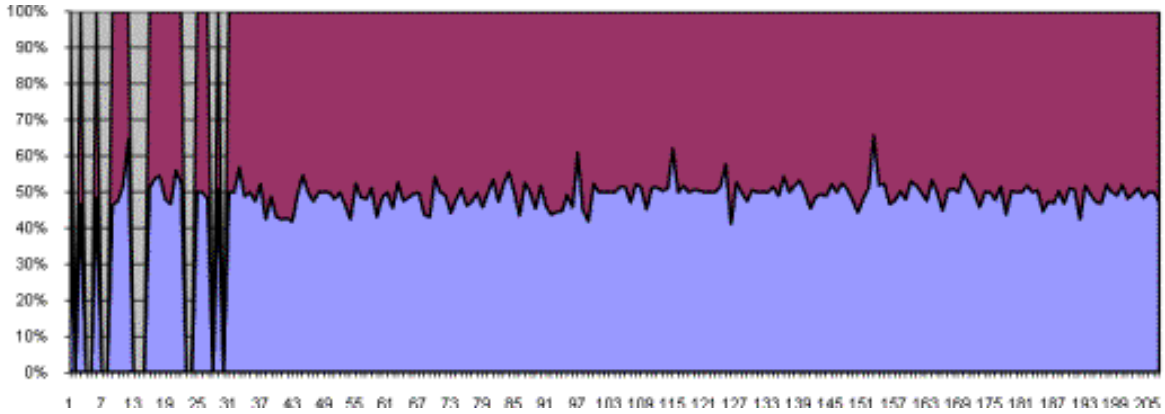

Figure 5.3 – Ratio of committed and delivered storypoints. If all committed storypoints are actually delivered a ratio of 1:1 is achieved placing the data point on the 50% (50:50) line.

(Note: Null data points have not been included in Fig5.3 as they introduce a distracting 'banding' effect. The banding effect shown in the figure is from data points with one or more data points missing.)

The relation between the number of storypoints committed to and delivered shows a high level of predictability – If a story is committed to there is a high probability that it will be delivered – and if not this will be communicated to the business as soon as this is known.

This ratio shows no obvious change over the period 2008 – 2009.

To attempt to identify a trend hidden in the noise due to different team requirements and uncalibrated storypoints these data were 'binned' into groups of 25, starting with the first 25 in 2008, until 2009. These data were aggregated and the ratio of committed storypoints to delivered storypoints calculated and plotted:

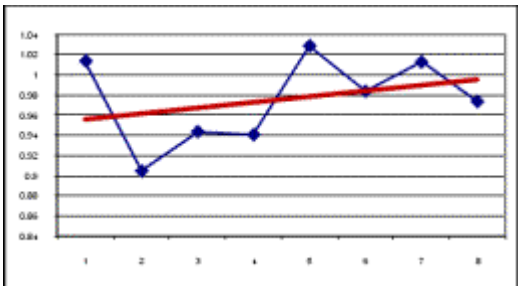

Figure 5.4 – Aggregated (smoothed) ratio of committed to delivered storypoints – in sequential groups for 25 from beginning of 2008 to August 2009

(Note 5: This is the wrong format for this graph. A bar chart would have been more appropriate. The line between points is meaningless.)

With the exception of the first set a trend of ratio delivered to committed emerges. Over time the ratio of those delivered by delivery teams increases and stabilizes at a level where what is committed to is delivered. An improvement emerges that was not discerned in the earlier analysis.

(It is not clear yet why the ratio of committed to delivered for first 25 releases (at the beginning of 2008) appears to be very high.)

During sprints additional stories can be added. The number of storypoints added to sprints during this time has decreased:

This analysis of scope predictability shows that as delivery date predictability improved so did the ability to deliver what was committed - there was no trade off (the usual option in strictly time boxed lifecycles, including 'traditional' Scrum).

Schedule and Scope Predictability

The two components of predictability have both been characterized. These both show significant improvement over the last two years.

Figure 5.8 summarizes these two aspects of predictability. Data points with a value of 1 have failed one of two delivery performance targets. Data points with a value of 2 have failed both.

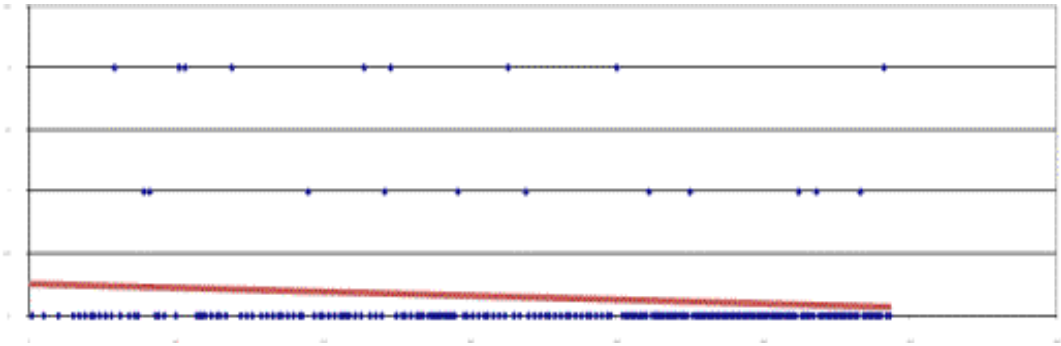

Figure 5.8 – Delivery performance indicators from January 2008 to August 2009. A delivery with a data point on level 1 indicates a failure of one key performance indicator. A delivery with a data point on level 2 indicates a failure to meet both key performance indicators.

It can be seen that over time the number of failures, in particular the number of 'double failures' decreases dramatically, as the number of deliveries meeting both performance targets increases.

Responsiveness, and Frequency and Regularity

Responsiveness is valued within ***bwin***. If changes to the commercial or legal environment occur changes must be made very quickly. ***Bwin*** has designed its delivery process to allow the introduction of new stories during a sprint.

Responsiveness is indicated by the number of stories introduced during sprints.

The graph below shows the number of storypoints (not stories) introduced during sprints during 2008 and 2009.

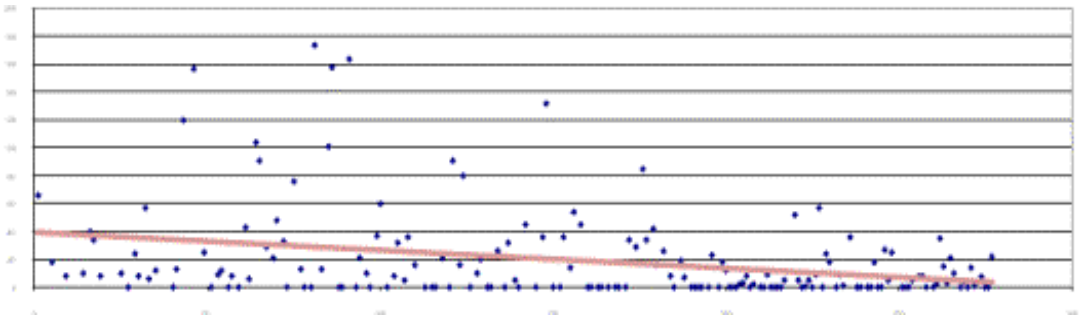

Figure 6.1 – Number of storypoints introduced during sprints from January 2008 through to August 2009.

It would appear that the number of storypoints introduced during a sprint has *declined* over this period *(suggesting a reduction in responsiveness)*..

This shows a reduction in the 'interrupt driven' approach to support. As requirements management has improved (as part of the DM process) and the DM process becomes better understood by delivery teams, dependent delivery teams and the

business the need to interrupt a sprint has declined as business and IT synchronize to the DM process 'rhythm'. However a residual level of genuine business driven interrupts remain as ***bwin*** responds to the commercial environment and legal requirements and delivery teams in turn respond by injecting stories mid sprint.

These data may also be tentatively interpreted as evidence for the emergence of a delivery 'rhythm', or the implementation 'carrier' within ***bwin***.

Quality

Bwin software must satisfy the stringent quality requirements: 250,000 concurrent users, multi-transactional, international (20 languages), real-time financial transactions, with over one million transactions per day

A useful proxy measure of the quality of the ***bwin*** systems supported by DM teams is the number of defects found in live operations.

The graph below show data extracted from AC Pro, the defect register. It shows the numbers of all defects reported in a calendar month (blue bars) and those reported from live operations in a calendar month (red bars).

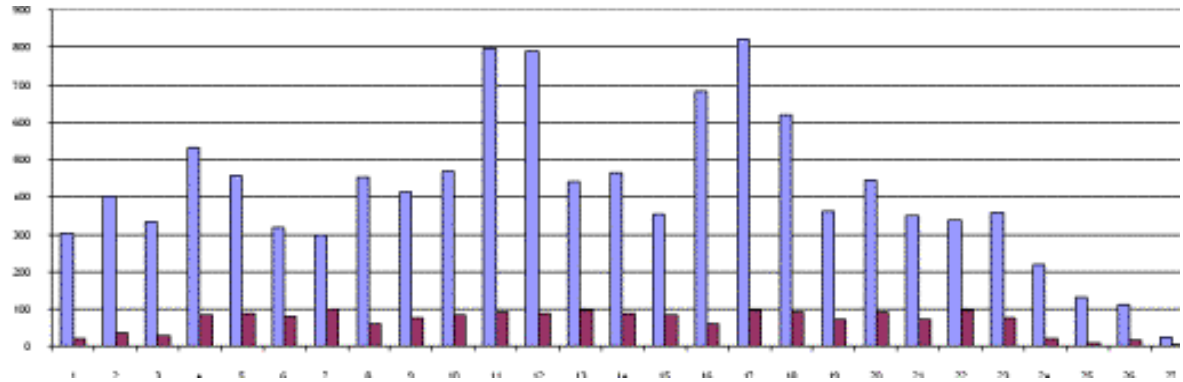

Figure 7.1 – Numbers of defects reported by month: Blue bars are all defects in a month; red bars are defects from live operation.

The number of defects reported from live operations remains consistent throughout the reporting period. This meets the ***bwin*** requirement for sustained software quality.

One third of the defects reported from live operations are high priority (about 500 of about 1800). With live defects being reported at about 60 to 100 per month this means about one high priority defect being reported per day.

Bwin software assets were about 400 KLOC (thousands of net, executable lines of code) in December 2007. This had increase to about 600 KLOC in December 2008 and to about 1 MLOC in September 2009.

This increase in LOC, together with consistent defect reporting levels suggests that, in practice, the quality of bwin software is improving.

In addition, if it is assumed that defects reported are mostly a result of new or changed code then the level of defects reported implies exceptional levels of code quality:

200 KLOC of new code (plus changed lines of code) in 2008 gave rise to about 1000 defect reports that year, of which about 400 where high priority. **This gives a defect density of about 5 defects/KLOC**.

Note: The data indicate that quality has been sustained (and improved) over this period and has not been 'traded off' against improved delivery predictability.

(Note 6: A more complete explanation (or justification!) of these data is available from the author upon request.)

Supporting Data

Data presented in the sections above provide a quantitative snapshot of improvement work at ***bwin***. Key attributes of the development work were identified, measures defined and targets set and met. Other improvements have been made too.

However improvements also manifest in other ways. The following remarks and comments from retrospectives and conversations with delivery process users capture co some of the other effects of this process improvement work:

- "Quite accurate detailed estimation
- "Good testing
- "Sensitive improvement of the code quality
- " '*VersionOne'* - big improvement in sprint tracking and communication"

- "Communication good via video-conference"
- "We have delivered everything we have committed"
- "Good collaboration between devs and testers"
- "Developer and tester worked closer this sprint"
- "Task breakdown well done"
- "Code quality considerably improved"
- "Not pressure on testing in the end of the sprint"
- "The sprint planning and commitment was well done for this sprint by removing from it not the lowest priority stories but those which could create an overload for George and Georg"
- "First time a best effort story was started"
- "Number of bugs significant reduced"
- "The development went quite fast this sprint."
- "Testing very good"

Other noteworthy observations:

- "Number of requestors has reduced (for instance, for one product, we went from 22 people generating requirements in 2007 to 1 person in 2009), allowing a controlled management of the requirements"
- "It now takes only weeks to set up a sprint or a team; previously it took several months"
- "The process was implicit because we all knew it – then we had rapid growth, so development started slowing down; then "agile" was implemented and helped – we did not know what we are doing, just applying the names. Creating stories was done from the outside, what was in the stories was not good; sprints are [now] a lot more predictable in most domains."
- "System is more flexible today than five years ago – this is required in order to respond to local markets and open up to new business"
- "Predictability and explicitness is better than previously"
- "Distributed development teams (between Austria, Poland, Romania and Sweden) can now be implemented because of the shared approach"
- "A single distribution manager can now manage two or three teams or sprints at the same time; previously we needed one manager per sprint"

The *bwin* Delivery Management Process

Since 2007 delivery teams, the R2D (Requirements to Delivery) project, the tools group, the recently formed CPI (continuous process improvement) Group, and many others have been involved in this process improvement work. They have worked to accelerate and direct the development of their working practices from tacitly understood agile development, adapted to the bwin business environment, to a well understood, well described common process, with well defined communication and control interfaces to the business and other development teams.

After some false starts this process was developed collaboratively to satisfy the business drivers by providing:

- effective and precise product backlog management
- careful estimation and collaborative planning
- continuous visibility – with early warning of arising problems
- integration of development and testing – and early test involvement

This common process was then provided with tool support and automation were possible. Which in turn enable the process to be 'instrumented' providing us with data for monitoring and control – some of which has been presented above to show how we have improved.

Our process improvement work, to improve delivery predictability and inter team dependencies, has transformed our agile development practice from a useful tactic into a novel strategic business capability

---

[1] As described and agreed locally within delivery teams

[2] About 60% of all deliveries are dependent on delivery by a previous - or, on occasion – concurrent delivery. capability.

5 Closing Remarks

The work undertaken by bwin to equip itself with a software development capability to meet its business needs has delivered an important and unique software development capability (in the author's experience).

This capability gives bwin:

1. a delivery process incorporating careful estimation and dependency management resulting in remarkable predictability, that in turn enables...

2. ...a scalable delivery process. Delivery teams working with this common process, and to a shared delivery rhythm, and with high visibility across teams collaborate easily and naturally, to deliver major business changes, fast.

3. While sustaining, and improving systems quality

A number of important points have emerged during the course of this work , concerning a) the ability to successfully transform a software development capability, b) the capability itself, and c) the ability to analyse and communicate information about that capability.

1) Agile development alone ('pure' agile development) while having some useful properties is not, necessarily, sufficient to meet business requirements.

2) Ideas about agile development have little useful to say about how agile teams should or could work collaboratively (beyond SoS).

3) Successful transformation of an agile capability (or any a capability for that matter) requires a clear idea of what, specifically, is required of that software development capability.

4) The willingness and ability to select, adopt and adapt ideas from agile practice, from, 'process oriented' software development and other 'reference models' (whether or not they appear to contradict or conflict) and to develop these into a coherent business specific 'operational model' is a major contributor to delivery of an effective software development capability (rather than simple imitation or mimicking of reference models).

5) Collaborative development by developers and technical staff, from the existing process baselines (rather than experts or consultants introducing or rolling out a new process) is a major factor in the success of this work.

6) Good process infrastructure (including

7) carefully selected and tailored tools) is an essential enabler of agile development practices and the achievement of required development capability. When these support a common agile process, across teams important new properties can emerge. (It could be argued that agile development is a result of good tools - in particular configuration management tools - enabling continuous integration and frequent builds.

8) The tools and infrastructure required to support a contemporary software development environment capture, by default, large quantities of day to day operational data concerning diverse aspects of the work and systems being developed. This data having served its purpose tends to be forgotten, but can, if carefully, but simply, analysed reveal patterns or trends that reflect the capability of the organization as well as suggesting areas for further, ongoing refinement.

An electronic copy is available at www.osel.co.uk/tesassc.pdf. Comments and correction to this draft paper would be welcomed by the author.

6 Acknowledgements

All of bwin's software managers and technical staff have contributed to the development of their remarkable software development capability. The author is grateful to them for their patience and perseverance in allowing their work to be investigated and reported.