An Agile Project Startup Week: 'Evo Start'

by Tom & Kai Gilb

We would like to describe how we start up agile projects, which are completed using our 'Evo' [6] agile method [2, 3].

We have been using exactly this Project start-up method world-wide, in many companies, and for both software/IT projects and other systems engineering projects (like 25 (now) Boeing Aircraft Projects in 1990) for decades, and it works. It gives a flying start to the incremental value delivery process; starting with value delivery, the 2nd week.

This process is appropriate for any consequent agile process, such as our 'Evo', which is focussed on delivering real measurable stakeholder value incrementally, as opposed to the majority of current agile methods which are focussed on delivering code; but, which do not attempt to define or deliver real stakeholder value itself, directly.

One solution to the agile problem of 'code fixation', which one of our multinational bank clients has recently adopted, for the wide variety of agile methods being used in the bank, is to suggest that the 'Evo' process [2] be *added* on top of their current agile process, for example on Scrum or/and XP. Evo then manages the stakeholder value, and Evo provides value design ideas to the code development team.

Evo will not only output ideas for code (a burn down stack), but will in fact output *any* (non code) design ideas that will deliver stakeholder value, such as training programmes, database construction, or motivational tactics. Evo operates at the systems engineering level, as Scrum allows in principle.

The Evo startup week is a sort of feasibility study, in the sense of

- Day 1: Drafting a feasible set of top 10 quantified project value objectives

- Day 2: Drafting a top 10 architecture hypothesis set

- Day 3: Estimating the multiple effects of all architecture on all value objectives, and critical resource constraints (budget, deadline)

- Day 4: Suggesting initial value delivery steps, next week

- Day 5: Getting management approval to proceed with the second week, and to see if we can really deliver value to stakeholders.

The Evo week is intentionally time boxed (one week), no matter what the *size of the project*. This is done so that:

- We do not get into weeks and months of bureaucratic start up overhead, before we have to deliver real value to stakeholders

- We will focus on the critical top level objectives [5]

- The detailed design will emerge iteratively, as a result of value measurement, and feedback.
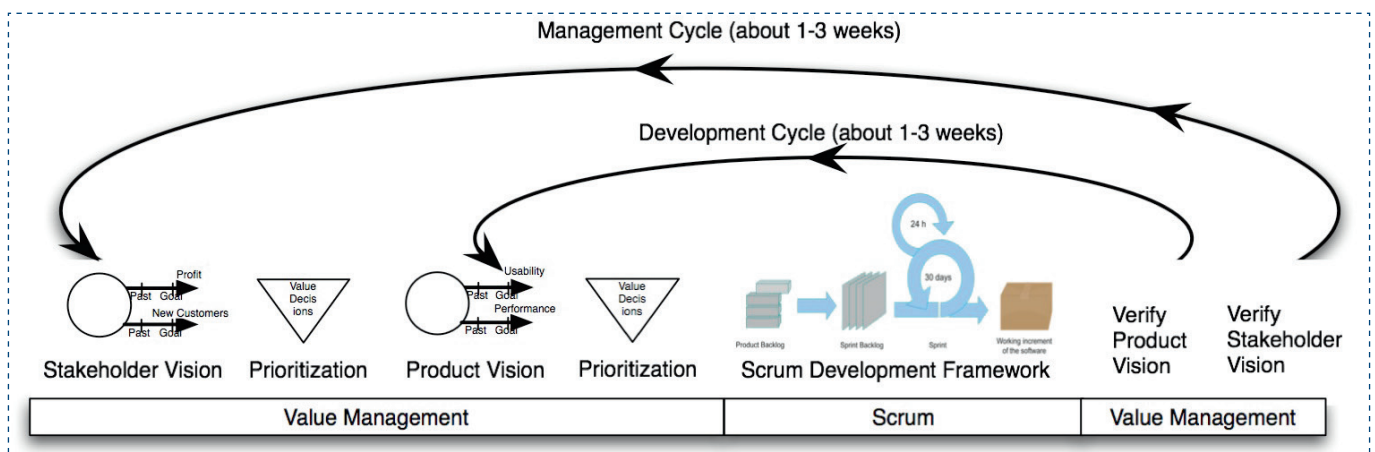

Figure 1: Two levels of result management, above a Scrum process. The 'Business level', on top of the stakeholder level is missing from this illustration here.

In practice, we gather a small group, a meeting room full, 6 to 14 people, for a week. Some specialist individuals can come and go during the week. When Kai and I do this with a client, we act as coaches. This week is also the training course in the Evo method, and this practical approach is far better than a week of classroom training, It is politically easier to budget than training, since it is 'real work on real projects'.

Day 1: Top 10 Critical Objectives, Quantified

Real Bank Project: Project Progress Testability
Quantification of the most-critical project objectives on day 1

P&L-Consistency&T P&L: Scale: total adjustments btw Flash/Predict and Actual (T+1) signed off P&L. per day. **Past 60 Goal: 15**

Speed-To-Deliver: Scale: average Calendar days needed from New Idea Approved until Idea Operational, for given Tasks, on given Markets.
Past [2009, Market = EURex, Task =Bond Execution] **2–3 months?**
Goal [Deadline =End 20xx, Market = EURex, Task =Bond Execution] **5 days**

Operational-Control: Scale: % of trades per day, where the calculated economic difference between OUR CO and Marketplace/Clients, is less than "1 Yen"(or equivalent).
Past [April 20xx] **10 %** change this to 90 % NH **Goal** [Dec. 20xy] **100 %**

Operational-Control.Consistent: Scale: % of defined [Trades] failing full STP across the transaction cycle. **Past** [April 20xx, Trades=Voice Trades] **95 %**
Past [April 20xx, Trades=eTrades] **93 %**
Goal [April 20xz, Trades=Voice Trades] **<95 ± 2 %>**
Goal [April 20xz, Trades=eTrades] **98.5 ± 0.5 %**

Operational-Control.Timely. End&OvernightP&L: Scale: number of times, per quarter, the P&L information is not delivered timely to the defined [Bach-Run].
Past [April 20xx, Batch-Run=Overnight] 1
Goal [Dec. 20xy, Batch-Run=Overnight] **<0.5>** **Past** [April 20xx, Batch-Run= **T+1**] **1 Goal** [Dec. 20xy, Batch-Run=End-Of-Day, Delay<1hour] **1**

Operational-Control.Timely.IntradayP&L: Scale: number of times per day the intraday P&L process is delayed more than 0.5 sec.

Operational-Control.Timely.Trade-Bookings: Scale: number of trades per day that are not booked on trade date. **Past** [April 20xx] **20?**

Front-Office-Trade-Management-Efficiency: Scale: Time from Ticket Launch to trade updating real-time risk view
Past [20xx, Function = Risk Mgt, Region = Global] **~80s ± 45s ??**
Goal [End 20xz, Function = Risk Mgt, Region = Global] **~50 % better?**
Managing Risk – Accurate – Consolidated – Real Time

Risk.Cross-Product: Scale: % of financial products that risk metrics can be displayed in a single position blotter in a way appropriate for the trader (i.e. – around a benchmark vs. across the curve).
Past [April 20xx] **0 %** 95 %. **Goal** [Dec. 20xy] **100 %**

Risk.Low-latency: Scale: number of times per day the intraday risk metrics is delayed by more than 0.5 sec. **Past** [April 20xx, NA] **1 %** **Past** [April 20xx, EMEA] **?? %** **Past** [April 20xx, AP] 100 % **Goal** [Dec. 20xy] **0 %** Risk.Accuracy

Risk. user-configurable: Scale: ??? pretty binary – feature is there or not – how do we represent?
Past [April 20xx] **1 %** **Goal** [Dec. 20xy] **0 %**

Operational Cost Efficiency: Scale: <Increased efficiency (Straight through processing STP Rates)>

Cost-Per-Trade: Scale: % reduction in Cost-Per-Trade
Goal (EOY 20xy, cost type = I 1 – REGION = ALL) **Reduce cost by 60 %** (BW)
Goal (EOY 20xy, cost type = I 2 – REGION = ALL) **Reduce cost by x %**
Goal (EOY 20xy, cost type = E1 – REGION = ALL) **Reduce cost by x %**
Goal (EOY 20xy, cost type = E 2 – REGION = ALL) **Reduce cost by 100 %**
Goal (EOY 20xy, cost type = E 3 – REGION = ALL) **Reduce cost by x %**

Figure 2. One page project requirements quantified.

Figure 2 shows the real (but modified for confidentiality) top ten objectives on a page. This is an extract, to fit on a page, of the larger detail we developed for each requirement. The requirements were put into a spread sheet tool developed by Kai Gilb, which supports the next 3 days (strategies, Impact Estimation, Selection of best next steps). The objectives are specified in Planguage [8], numerically.

In practice, when 10 objectives need defining in a single day, we split up the work in parallel teams, who do a few objectives each, and put them together by end of day.

Day 2. The Strategies and Architcture: Top 10 Means to our ends

The second day, our team focuses on identifying what we believe (a 'hypothesis' to be proven, or not) are the up to 10 most important

things to implement, in order to reach our top 10 objectives. These are things like, build from scratch, reuse a set of current system components, and use architecture we have patents or experience for. Initially we will just *name* the strategies. If they are already well known and well documented ideas (like, 'our old interfaces', or Product X), we do not need to detail them. We should have enough information to estimate their effects and costs (Day 3). But if the architecture/strategy ideas are being created in detail by us, then we need to specify them in some detail, in order to have a basis for understanding their effects on our top 10 objectives, and on our costs.

Defining a Design/Solution/Architecture/Strategy (Planguage, CE Design Template)
1. enough detail to estimate, 2. some impact assertion, 3. Assumptions, Risks, Issues

Orbit Application Base: (formal Cross reference Tag)
Type: Primary Architecture Option
============= Basic Information ==========
Version: Nov. 30 20xx 16:49, updated 2.Dec by telephone and in meeting. 14:34
Status: Draft
Owner: Brent Barclays
Expert: Raj Shell, London
Authority: for differentiating business environment characteristics, Raj Shell, Brent Barclays (for overview)
Source: <Source references for the information in this specification. Could include people>. Various, can be done later BB
Gist: risk and P/L aggregation service, which also provides work flow/adjustment and outbound and inbound feed support. Currently used by Rates ExtraBusiness, Front Office and Middle Office, USA & UK.
Description: <Describe the design idea in sufficient detail to support the estimated impacts and costs given below>.
   D1: ETL Layer. Rules based highly configurable implementation of the ETL Pattern, which allows the data to be onboarded more quickly. Load and persist new data very quickly. With minimal development required → Business-Capability-Time-To-M Business Scalability
   D2: high performance risk an aggregation processing (Cube → Timeliness, P/L Explanatio P/L Understanding, Decision Business Scalability, Respons
   D3: Orbit supports BOTH Risk P/L Explanation, Risk & P/L C Risk & P/L Understanding, Decision Support.
   D4: a flexible configurable workflow tool, which can be used to easily define new workflow processes → Books/ Records Consistency, Business Process Effectiveness, Business Capability Time to Market.
   D5: a report definition language, which provides 90+% of the business logic contained with Orbit, allows a quick turnaround of new and enhanced reports with minimal regression testing and release procedure impact. → P/L Explanation, Risk & P/L Understanding, Business Capability Time to Market, Business Scalability.
   D6: Orbit GUI. Utilizes an Outlook Explorer metaphor for ease of use, and the Dxx Express Grid Control, to provide high performance Cube Interrogation Capability. → Responsiveness, People Interchangeability, Decision Support, Risk & P/L Understanding.
   D7: downstream feeds. A configurable event-driven data export service, which is used to generate feeds. → Business Process Effectiveness, Business Capability Time to Market.

======= Priority and Risk Management =======
Assumptions: <Any assumptions that have been made>.
   A1: **FCCP is assumed to be a part of Orbit.** FCxx does not currently exist and is Dec 20xx 6 months into Requirements Spec. ← Picked up by TsG from dec 2 discussions AH MA JH EC.
      Consequence: FCxx must be a part of the impact estimation and costs rating.
   A2: **Costs,** the development costs will not be different. All will base on a budget of say $nn mm and 3 years. The o+ costs may differ slightly, like $n mm for hardware. MA AH 3 dec
   A3:Boss X will continue to own Orbit. TSG DEC 2
   A4: the schedule, 3 years, will constrained to a scope we can in fact deliver, OR we will be given additional budget. If not "I would have a problem" ← BB
   A5: the cost of expanding Orbit will not be prohibitive. ← BB 2 dec
   A6: we have made the assumption that we can integrate Oribit with PX+ in a sensible way, even in the short term ← BB
Dependencies: <State any dependencies for gn idea>.
   replaces Px+ in time. ? tsg 2.12 e or refer to tags of any factors, uld threaten your estimated .
   is delayed. Mitigation: continue x ← tsg 2.12
   echnical **integration** of Px+ easy as thought & we must redevelop Orbit
   R3: the and or scalability and cost of **coherence** will not allow us to meet the delivery.
   R4: **scalability** of Orbit team and infrastructure, first year especially ← BB. People, environments, etc.
   R5: re Cross Desk reporting Requirement, major impact on technical design. **Solution not currently known.** Risk no solution allowing us to report all P/L
Issues: <Unresolved concerns or problems in the specification or the system>.
   I1: Do we need to put the fact that we own Orbit into the objectives (Ownership). MA said, other agreed this is a huge differentiator. Dec 2.
   I2: what are the time scales and scope now? Unclear now BB
   I3: what will the success factors be? We don't know what we are actually being asked to do. BB 2 dec 20xx
   I4: for the business other than flow options, there is still a lack of clarity as to what the requirements are and how they might differ from Extra and Flow Options. BB
   I5: the degree to which this option will be seen to be useful without Intra Day. BB 2 dec

Figure 3. One-page plan.

Figure 3 is a one-page detailed description of a single top ten strategy. We did this by telephone interview with the strategy expert (he was snowed out of travelling to work that day), in one hour. We used the Design Template in the Design Chapter of the Competitive Engineering book. But we did not yet make impact estimates, which are in that template. Those will be done day 3, and integrated into an Excel Tool (Kai Gilb) with all other specifications and estimations. It should be obvious that detailing the design, and collecting info on risks, dependencies, issues, and assumptions is valuable and necessary for realistic understanding of main objective effects, and costs, of the strategy.

Day 3: Impact Estimation; estimating the value for our objectives, and the concurrent costs of the strategies

On the third day, the team attempts a complete estimation of the main and side effects, and the costs of all strategies, on all main objectives. For a 10×10 table that is 100 estimates, plus maybe 2×10 = 20 cost estimates. In addition to the estimates, if we have time, and want a high quality auditable architecture hypothesis, we also add to each estimate:

- the ± uncertainty, the range, of our estimate. The spread of experience
- the evidence, facts, actual measures, measuring processes and sources

Based on the above we designate the Credibility of each estimate on a scale of 0.0 (none) to 1.0 (perfect credibility)

We use the ± uncertainty, and the credibility index to modify our basic estimates, in the direction of more conservative estimates. We calculate a 'worst, worst case'.

Strategies / Goals	Identify Binding Compliance Requirements Strategy	System Control Strategy	System Implementation Strategy	Find Services That Meet Our Goals Strategy	Use The Lowest Cost Provider Strategy
Security Administration Compliance 25 % → 90 %	100 %	100 %	100 %	50 %	0 %
Security Administration Performance 24 hrs → 4 hrs	75 %	100 %	100 %	100 %	0 %
Security Administration Availability 10 hrs → 24 hrs	0 %	0 %	0 %	100 %	0 %
Security Administration Cost 100 % → 60 %	50 %	100 %	100 %	100 %	100 %
Total Percentage Impact	225 %	300 %	300 %	350 %	100 %
Evidence	ISAG Gap Analysis Oct. 03	John Collins	John Collins	John Collins	John Collins
Cost to Implement Strategy	15 man days (US$ 5,550)	15 man days (US$ 5,550)	15 man days (US$ 5,550)	15 man days (US$ 5,550)	1 man day (US$ 1,110)
Credibility	0.9	0.6	0.6	0.75	0.9
Cost Adjusted Percentage Impact	202.5 %	180 %	180 %	262.5 %	90 %

Figure 4. Acer Project: Impact Estimation Table.

Figure 4, a real Impact estimation table. The '100 % → 60 %' expressions on the left are a reference to the Past levels (0 % impact) and the Goal levels (100 %) of defined Scale objectives. This is of course a different project than the one in examples above. But it was less complicated, so we chose it as an illustration.

Day 3, estimation makes us think deeply about what we have proposed, and what we really know about the ideas. It is impressive to witness the advanced level of fact-based logical discussion amongst participants.

Day 4: Extract a Value Delivery Step for Next Week

The 4th day is about finding something practical to do the next week (and ultimately each week thereafter, until all Goal levels are reached). This applies even when the current system is ultimately going to be scrapped. But we will usually install new improvements on the current system initially, in order to get some real improvements quickly. There is no question of building a whole new system, next week, but there is always (believe it or not) something practical we can do to start the process of testing the architectural hypothesis, to start pleasing stakeholders, to start building our credibility and confidence. The Impact estimation table has a strong clue as to where to find high value increments. And most people (some are 'in denial') can think of several practical small steps to get us going-

Day 5: Getting approval to start rolling out the value

The 5th day, sometimes done at end of 4th, or integrated into the other 4 days, if the executive participates actively, is simply getting a 'go ahead next week, at least' signal from a responsible manager. This is easy because the lure of real results in the short term is convincing. What can you lose?

Using this process in agile IT projects, such as in Banking, is our norm. But the process seems to work in any planning area. In 1990 we used it on 25 aircraft projects at McDonnell-Douglas. We did 5 parallel start-ups, in 5 different weeks, and they all got approved. One device we used there, was to involve the executive for half an hour every evening. We got daily buy in, correction, and understanding by management. No surprises on Friday.

People learned the agile method, Evo, by *doing*, rather than any classroom training. And we could budget the exercise as part of the project, not as a training overhead.

If you want to try Evo, and Evo startups, be sure to download the free standards referenced here [1, 2] as a guide. Let us know how this works for you in your agile environment. Or, write a paper for Agile Record, or their Conferences.

References

[1] Evo Startup Standard, Jan 12 2013
     http://www.gilb.com/dl562

[2] Evo Project Management Standard, Jan 12 2013
     http://www.gilb.com/dl563

[3] This is a detailed standard for conducting an 'Evo' (Evo-
     lutionary Project Management, Gilb's Agile Method) as
     described in my book Competitive Engineering, Chapter 10
     [http://www.gilb.com//tiki-download_file.php?fileId=77]

[4] The slides increments of value delivery (10 min slides)
     [http://www.gilb.com/tiki-download_file.php?fileId=451]
     give a case study view of using this method for DoD. Pers-
     inscom, US Army IT System.

[5] "The Top 10 Critical Requirements are the Most Agile Way
     to Run Agile Projects"
     in Agile Record, August 2012, Issue 11, pages 17, 19-21,
     http://www.gilb.com/dl554

[6] The 'Evo' abbreviation was probably first adopted for the
     use of our Evolutionary Project Management methods in
     cooperation with Hewlett Packard's widespread use of
     Evo, as reported in HP Journal, for example Todd Cotton,
     "Evolutionary Fusion: A Customer- Oriented Incremental
     Life Cycle for Fusion", HP Journal August 1996. We taught
     the method to Todd about 1988 on an HP Project, and he
     was a champion of the Evo method at HP. http://www.hpl.
     hp.com/hpjournal/96aug/aug96a3.htm
     'Evo' is simply a abbreviation of 'Evolutionary'. Though we
     have played with acronyms like Evolutionary Value Optimi-
     zation.

[7] Experience in Multinational Banks, conference lecture
     slides. http://www.gilb.com/dl532. This contains several
     realistic case studies regarding Evo project startup. Testing
     and Finance Conf., London 2012

[8] Planguage is defined in Gilb, Competitive Engineering,
     2005.   ■

> about the authors

Tom Gilb and Kai Gilb

Tom Gilb and Kai Gilb have, together with many profes-
sional friends and clients, personally developed the Agile
methods they teach. The methods have been developed
over five decades of practice all over the world in both
small companies and projects, as well as in the largest
companies and projects. Their website **www.gilb.com/
downloads** offers free papers, slides, and cases about Agile and other subjects.
There are many organisations, and individuals, who use some or all of their
methods. IBM and HP were two early corporate-wide adopters (1980, 1988).
Recently (2012) over 15,000 engineers at Intel have voluntarily adopted the
Planguage requirements specification methods; in addition to practicing to a
lesser extent Evo, Spec QC and other Gilb methods. Many other multinationals
are in various phases of adopting and practicing the Gilb methods. Many
smaller companies also use the methods.

Tom Gilb

Tom is the author of nine published books, and hundreds of papers on Agile
and related subjects. His latest book 'Competitive Engineering' (CE) is a de-
tailed handbook on the standards for the 'Evo' (Evolutionary) Agile Method,
and also for Agile Spec QC. The CE book also, uniquely in the Agile community,
defines an Agile Planning Language, called 'Planguage' for Quality Value
Delivery Management. His 1988 book, Principles of Software Engineering
Management (now in 20th Printing) is the publicly acknowledged source of
inspiration from leaders in the Agile community (Beck, Highsmith, and many
more), regarding iterative and incremental development methods. Research
(Larman, Southampton University) has determined that Tom was the earliest
published source campaigning for Agile methods (Evo) for IT and Software.
His first 20-sprint agile (Evo) incremental value delivery project was done in
1960, in Oslo. Tom has guest lectured at universities all over UK, Europe,
China, India, USA, Korea – and has been a keynote speaker at dozens of
technical conferences internationally.

Kai Gilb

Kai Gilb has partnered with Tom in developing these ideas, holding courses
and practicing them with clients since 1992. He coaches managers and
product owners, writes papers, develops the courses, and is writing his own
book, 'Evo – Evolutionary Project Management & Product Development.'
Tom & Kai work well as a team; they approach the art of teaching their
common methods somewhat differently. Consequently the students benefit
from two different styles.