From Waterfall to Evolutionary Development (Evo): How we rapidly created faster, more user-friendly, and more productive software products for a competitive multi-national market

Trond Johansen
Head of Project Management, FIRM
Trond.Johansen@firmglobal.com

Tom Gilb
Tom@Gilb.com

Abstract. Evolutionary development (Evo) focuses on early delivery of high value to stakeholders, and on obtaining and utilizing feedback from stakeholders. This paper describes from a project manager's viewpoint, the positive experiences that one organization rapidly achieved on switching from using the Waterfall method to Evo. Major benefit came from paying greater attention to the quality requirements as opposed to the previous practice of concentrating solely on the required functionality.

INTRODUCTION

About the company

Future Information Research Management (FIRM) was established in 1996, and has 70 employees in 4 offices (Oslo, London, New York and San Francisco). FIRM delivers one software product, Confirmit, which is a web-based application that enables organizations to gather, analyze and report key business information across a broad range of commercial applications. Confirmit can be applied to any information-gathering scenario, but its three main data sources are Customer Feedback, Market Feedback and Employee Feedback.

The FIRM R&D department consists of about 20 people, including a Quality Assurance department of 3 people. These people are mainly involved in product development of Confirmit, but they also carry out some custom development for clients who fund new modules of the software.

Development background & history

In the very beginning, when FIRM only had a couple of clients, our development was very ad-hoc and customer driven. We didn't follow a formal development process. The software was updated nearly on a daily basis based on client feedback. You can say that we initially had one of the important elements of Evo: deliver stakeholder value *fast*.

This ad-hoc development resulted in nice features for the few dedicated clients we had, but it also resulted in a lot of defects, long stressful nights, and little control.

As our client base grew, we felt a need to introduce more-formal processes in order to increase our quality standards. Larger clients started to ask leading questions regarding our development processes.

We formalised the development process according to a Waterfall model, and started climbing the CMM ladder. The reason for choosing the Waterfall model was that it was the only development process we knew about.

After a few years with the Waterfall model, we experienced aspects of the model that we didn't like:

- Risk mitigation was postponed until late stages;
- Document-based verification was postponed until late stages;
- Attempts to stipulate unstable requirements too early: change of requirements is perceived as a bad thing in waterfall;
- Operational problems discovered too late in the process (acceptance testing);
- Lengthy modification cycles, and much rework;
- Most importantly, the requirements were nearly entirely focused on functionality,, not on quality attributes.

Others have reported similar experiences:

- In a study of failure factors in 1027 IT projects in the UK, *scope management* related to *Waterfall* practices was cited to be the largest problems in 82% of the projects. Only approximately 13% of the projects surveyed didn't fail (Taylor 2000);
- A large project study, Chaos 2000 by The Standish Group showed that 45% of requirements in early specifications were never used (Johnson 2002).

THE SHIFT OF FOCUS: FROM WATERFALL TO EVO

Peter Myklebust, FIRM Chief Technical Officer (CTO), and I heard Tom & Kai Gilb speak about evolutionary project management (Evo) at a software conference in Autumn 2003. We had just released a new version of our software that contained a lot of nice new features, but it had limitations with respect to usability and productivity (for example, throughput and response time). We found the ideas very interesting, and Tom and Kai Gilb offered to give a more detailed introduction to the concept. They spent one day in our offices, giving a very compressed introduction to Evo. We saw that Evo attacked many of the flaws in our Waterfall process; most importantly, it gave a high focus to quality attributes - which we felt could have been better in our latest release.

We decided to carry out an Evo pilot with a development phase of 3 months. We decided to conduct a literature study ourselves, and then use Evo as best as we could for the next release (Confirmit 8.5), without further Evo courses initially.

FIRM's interpretation of Evo: the basis for the 3-month trial period

In a nutshell, Evo involves 'quickly evolving towards stakeholder values and product qualities, while learning through early feedback'. The beauty lies with the simplicity of the method, combined with the advanced methods of measurement and control.

After the one-day crash course with Tom and Kai Gilb and carrying out a literature study (that is, reading "Competitive Engineering" by Tom Gilb (Gilb 2005), "Evo: Evolutionary Project Management & Product Development" by Kai Gilb, and other material on the subject), our overall understanding of the Evo process was as follows:

- Find stakeholders (for example, end users, support, sales, and IT operations);
- Define the stakeholders' real needs, and the related product qualities;
- Identify the past status of the product qualities, and determine your required goal level (that is, how much you want to improve);
- Identify possible solutions (also known as 'designs') for meeting your goal levels on time;
- Develop a step-by-step plan (one step at a time) for delivering improvements via the

identified solutions, with respect your requirements;

And most importantly:

- Deliver measurable stakeholder-valued results every Evo cycle (An Evo cycle being approximately 2% of project time);
- Measure each Evo cycle: Are we measurably moving towards our goals?

Working with requirements the Evo way

Using Evo, our requirements process changed. Previously, we focused mostly on the functional requirements, and not on the quality requirements. We realized it is the *quality* requirements that really separate us from our competitors. For example, take the spell checker in MS Word, why was this a 'killer application'? There was no new functionality; authors of documents have been able to spell check with paper dictionaries for ages. The real difference was the superior product qualities: *the speed of spell checking and the usability*.

We tried to define our requirements according to a basic standard (Gilb 2005) as follows:

- Clear & Unambiguous;
- Measurable;
- Testable;
- No Solutions (Designs);
- Stakeholder Focus.

Usability.Productivity:

Scale: Time in minutes to set up a typical specified Market Research report (MR).

Past: 65 minutes. **Tolerable**: 35 minutes. **Goal**: 25 minutes.

Note: The actual end result was 20 minutes!

Meter: *(How to measure if we are moving towards our goal)*: Candidates with knowledge of MR-specific reporting features performed a set of predefined steps to produce a standard MR report. (The standard MR report was designed by Mark Phillips, an MR specialist at our London office.)

Figure 1. This shows an example of a quality requirement for Confirmit 8.5. The focus is here on the day-to-day operations of our MR users, not a list of features that they might or might not like. We *know* that increased efficiency, which leads to more profit, will please them

After *one week* we had defined *nearly all* the top-level quality requirements for the next version of Confirmit; and we were ready to start on our first Evo step. We decided that one Evo step should last one week for practical reasons, even though we violated the general Evo policy of not spending more than *about* 2 % of project schedule in each step. The rationale behind the 2% rule is not to spend more time than you can afford to loose. After one week, you'll find out whether you are on the right track (by getting feedback from stakeholders).

Find Solutions that takes you closer to your goals

For every quality requirement, we looked for possible solutions ('design ideas'). For example, for the quality requirement: Usability.Productivity, we identified the following solutions: (*Identified here by their name, not their description*)

- Solution.Recoding;
- Solution.MRTotals;
- Solution.Categorizations;
- Solution.TripleS;
- and many more.

We evaluated all of these, and specified in more detail those we believed would add the most value (that is, take us closer to the goal level).

Working evolutionary, the FIRM Evo week

We organized the week in a special way. On Friday we *plan* deliverables for version N, and we build and deploy version N-1 on the test server. Monday to Thursday is dedicated to design, code and test. During the week, the project collects feedback from stakeholders, based on the *previous* Evo step/week (See Table 2).

	Development Team	**Users (PMT, Pros, Doc writer, other)**	**CTO (Sys Arch, Process Mgr)**	**QA (Configuration Manager & Test Manager)**
Friday	• PM: Send Version N detail plan to CTO + prior to Project Mgmt meeting • PM: Attend Project Mgmt meeting: 12.00-15.00 • Developers: Focus on genereal maintenance work, documentation.		• Approve/reject design & Step N • Attend Project Mgmt meeting: 12-15	• Run final build and create setup for Version N-1. • Install setup on test servers (external and internal) • Perform initial crash test and then release Version N-1
Monday	• Develop test code & code for Version N	• Use Version N-1		• Follow up CI • Review test plans, tests
Tuesday	• Develop Test Code & Code for Version N • Meet with users to Discuss Action Taken Regarding Feedback From Version N-1	• Meet with developers to give Feedback and Discuss Action Taken from previous actions	• System Architect to review code and test code	• Follow up CI • Review test plans, tests
Wednesday	• Develop test code & code for Version N			• Review test plans, tests • Follow up CI
Thursday	• Complete Test Code & Code for Version N • Complete GUI tests for Version N-2			• Review test plans, tests • Follow up CI

Table 1: The FIRM Evo week

Evolutionary project planning

We collected the most promising solutions and included them in an Evo plan, which was expressed by using an Impact Estimation (IE) table (See Tables 2 and 3). The solutions were evaluated with respect to *value for clients* versus *cost of implementation*. We chose the ones with the highest value first. Note that value can sometimes be defined as *removing risks* by implementing technically challenging solutions early.

The IE table is our tool for controlling the qualities, and delivering improvements to real stakeholders, or as close as we can get to them (for example, delivering to our support people, who use the system daily and so can act as 'clients').

Recoding:

Type: Solution [Confirmit 8.5].

Description: Make it possible to recode a variable, on the fly, from Reportal.

Estimated effort: 4 days.

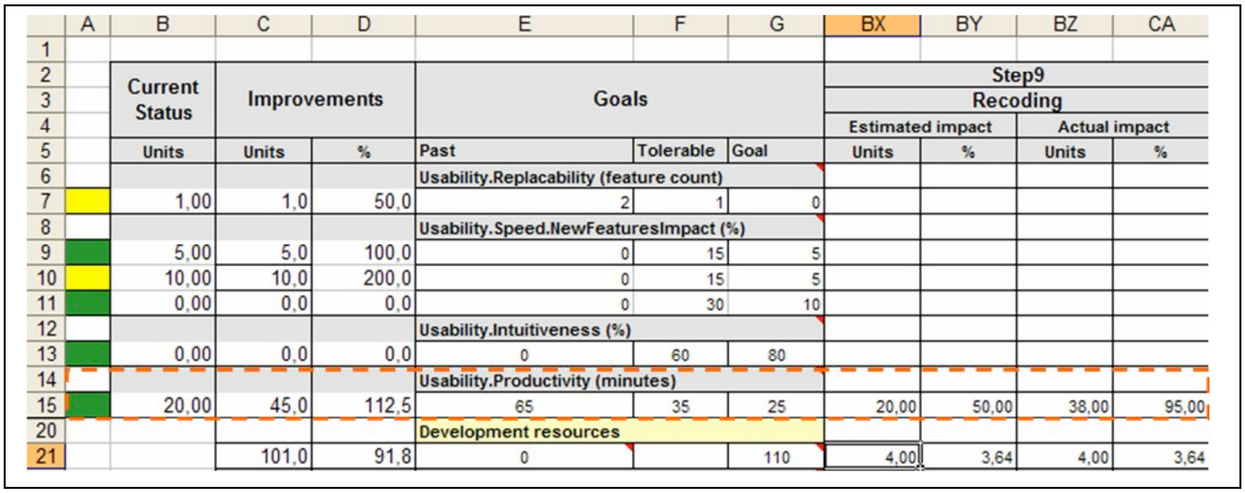
Figure 2. A brief specification of the solution, 'Recoding'

	Design Idea: Step 9 - Recoding			
Requirements	**Estimated Scale Impact**	**Estimated % Impact**	**Actual Scale Impact**	**Actual % Impact**
Objectives				
Usability.Productivity 65 <-> 25 minutes Past: 65 minutes. Tolerable: 35 minutes. Goal: 25 minutes.	**65 – 20 = 45 minutes**	**50%**	**65 - 38 = 27 minutes**	**95%**
Resources				
Development Cost 0 <-> 110 days	**4 days**	**3.64%**	**4 days**	**3.64%**

Table 2: Here is a simplified version of the IE table for Evo Step 9, 'Recoding' of the MR project. Notice the definitions for the requirements and costs. The Planguage keyed icon '<->' means 'from baseline to target value. Step 9 alone moved the Productivity value to 27 minutes, or 95% of the way to the target level

The Evo method's impact on Confirmit's product qualities

The Evo method's impact on Confirmit's product qualities is *not* actually measured statistically, by doing a scientifically correct large-scale survey, although we are currently considering this. The impact described in this paper for Confirmit 8.0 ('Past') is based on internal usability tests, productivity tests, performance tests carried out at Microsoft Windows ISV laboratory in Redmond USA, and from direct customer feedback. See Table 1 - only highlights of the impacts are listed here. No negative impacts are hidden.



	A	B	C	D	E	F	G	BX	BY	BZ	CA
1											
2		Current Status	Improvements		Goals			Step9			
3								Recoding			
4								Estimated impact		Actual impact	
5		Units	Units	%	Past	Tolerable	Goal	Units	%	Units	%
6					Usability.Replacability (feature count)						
7	🟨	1,00	1,0	50,0	2	1	0				
8					Usability.Speed.NewFeaturesImpact (%)						
9	🟩	5,00	5,0	100,0	0	15	5				
10	🟨	10,00	10,0	200,0	0	15	5				
11	🟩	0,00	0,0	0,0	0	30	10				
12					Usability.Intuitiveness (%)						
13	🟩	0,00	0,0	0,0	0	60	80				
14					Usability.Productivity (minutes)						
15	🟩	20,00	45,0	112,5	65	35	25	20,00	50,00	38,00	95,00
20					Development resources						
21			101,0	91,8	0		110	4,00	3,64	4,00	3,64

Table 3: Details of the real table, which was simplified in Table 2

Description of requirement/work task	Past	Status
Usability.Productivity: Time for the system to generate a survey	7200 sec	15 sec
Usability.Productivity: Time to set up a typical specified Market Research-report (MR)	65 min	20 min
Usability.Productivity: Time to grant a set of End-users access to a Report set and distribute report login info.	80 min	5 min
Usability.Intuitiveness: The time in minutes it takes a medium experienced programmer to define a complete and correct data transfer definition with Confirmit Web Services without any user documentation or any other aid	15 min	5 min
Workload Capacity.Runtime.Concurrency: Maximum number of simultaneous respondents executing a survey with a click rate of 20 seconds and an response time < 500 milliseconds, given a defined [Survey-Complexity] and a defined [Server Configuration, Typical].	250 users	6000

Table 4: Improvements to product qualities in Confirmit 8.5

These leaps in product qualities would not have been achieved without Evo. We have received many pleasant emails regarding these quality improvements from our customers. To give an example:

"I just wanted to let you know how appreciative we are of the new "entire report" export functionality you recently incorporated into the Reportal. It produces a fantastic looking report, and the table of contents is a wonderful feature. It is also a HUGE time saver."

On the *second* Release, Confirmit 9.0, the Vice President (VP) of Marketing proudly named the Evo development method on the FIRM website[1], and the quantified results they can deliver to

[1] I invite the reader to imagine a real marketing VP saying "We just started using [CMMI/RUP] 3 months ago and I want to report some fantastic quantitative product improvements"…. ☺

customers. I cannot recall a technical paper about development methods with anything similar - a line executive bragging about a development method!

> "FIRM, through evolutionary development, is able to substantially increase customer value by focusing on key product qualities important for clients and by continuously asking for their feedback throughout the development period. Confirmit is used by the leading market research agencies worldwide and Global 1000 companies, and together, we have defined the future of online surveying and reporting, represented with the Confirmit 9.0."

Figure 3. Comments by FIRM's VP of Marketing, Kjell Øksendal

The *above* quote by the VP of Marketing summarizes executive satisfaction with the new development method on their website. The detail below is from the same website product release

News release

2004-11-29: Press Release from FIRM

New version of Confirmit increases user productivity up to 80 percent

NOVEMBER 29th, 2004: FIRM, the world's leading provider of online survey & reporting software, today announced the release of a new version of Confirmit delivering substantial value to customers including increased user productivity of up to 80 percent.

FIRM is using Evolutionary (EVO) development to ensure the highest focus on customer value through early and continuous feedback from stakeholders. A key component of EVO is measuring the effect new and improved product qualities have on customer value. Increased customer value in Confirmit 9.0 includes:

* Up to 175 percent more intuitive user interface*
* Up to 80 percent increased user productivity in questionnaire design and testing*
* Up to 1500 percent increased performance in Reportal and Panel Management*

Figure 4. Confirmit 9.0 release announcement from the FIRM website, http://www.firmglobal.com. It gives detail about the method and the quantified product results

See Tables 5, 6, 7 and 8 for some more detailed results from Confirmit 9.0, by product component.

Feedback from developers and project managers within FIRM R&D

Evo has resulted in increased *motivation* and *enthusiasm* amongst developers, because it opens up for *empowered creativity*. This means that the developers can determine their own design ideas, and are not subject to being dictated the design ideas by marketing and/or customers, who often tend to be amateur technical designers.

Daily product builds were introduced, which we called Continuous Integration (CI). Evo combined with CI, is a vehicle for innovation and inspiration. The developers get their work out onto the test servers, and receive feedback. Every week.

Authoring		
Product Quality	**Description**	**Customer Value**
Intuitiveness	Probability that an inexperienced user can intuitively figure out how to set up a defined Simple Survey correctly.	Probability increased by 175%
Productivity	Time in minutes for a defined advanced user, with full knowledge of 9.0 functionality, to set up a defined advanced survey correctly.	Time reduced by 38%

Reportal		
Product Quality	**Description**	**Customer Value**
Performance	Number of responses a database can contain if the generation of a defined table should be run in 5 seconds.	Number of responses increased by 1400%

Survey Engine		
Product Quality	**Description**	**Customer Value**
Productivity	Time in minutes to test a defined survey and identify 4 inserted script errors, starting from when the questionnaire is finished to the time testing is complete and is ready for production. (Defined Survey: Complex survey, 60 questions, comprehensive JScripting.)	Time reduced by 83% and error tracking increased by 25%

Panel Management		
Product Quality	**Description**	**Customer Value**
Performance	Maximum number of panelists that the system can support without exceeding a defined time for the defined task, with all components of the panel system performing acceptably.	Number of panelists increased by 1500%
Scalability	Ability to accomplish a bulk-update of X panelists within a timeframe of Z seconds.	Number of panelists increased by 700%
Intuitiveness	Probability that a defined inexperienced user can intuitively figure out how to do a defined set of tasks correctly.	Probability increased by 130%

Tables 5, 6, 7 and 8: The tables show some detailed results from Confirmit 9.0, by product component

Even though they embraced the method, there are parts of Evo that they found difficult to understand and execute at first. These included:

- Defining good requirements can be hard;
- It was hard to find meters (that is, ways of measuring numeric qualities, testing quality levels), which were practical to use, and at the same time measured real product qualities;
- Sometimes it takes more than a week to deliver something of value to the client;
- Testing was sometimes postponed in order to start the next step. Some of these test deferments were then not in fact done in later testing.

Lessons learned with respect to the method

Some of the lessons we learned after the trial period include:

- We need to have increased focus on feedback from clients. We will select the ones that are willing to dedicate time to us. *Internal* stakeholders (like sales and help desk staff) can give valuable feedback, but some *customer* interaction is necessary;
- Demonstrate new functionality automatically, with screen recording software or early test plans. This makes it easier for internal and external stakeholders to do early testing;
- Tighter integration between Evo and the test process is necessary;
- 'Be humble in your promises, but overwhelming in your delivery'.

CONCLUSIONS

The Evo method's positive impact on Confirmit product qualities has convinced us that Evo is a better suited development process than our former Waterfall process, and we will continue to use Evo in the future.

What surprised us the most was the method's power of focusing on delivering value for clients versus the cost of implementation. Evo enables you to re-prioritize the next development-steps based on the weekly feedback. What seemed important at the start of the project may be replaced by other solutions based on gained knowledge from previous steps.

The Evo method has high focus on measurable product qualities, and defining these clearly and testably requires training and maturity. It is important to *believe* that everything can be measured and to seek guidance if it seems impossible.

One pre-requisite related to the method for using Evo is an *open architecture*.

Another pre-requisite is *management support* for changing the work process, and this is important in any software process improvement initiative.

The concept of daily builds, CI, was valuable with respect to delivering a new version of the software every week.

Overall, the whole organization has embraced Evo. The release of Confirmit 8.5 showed some of Evo's great potential, and we will work hard to utilize it to the full in the future. In June 2004, we had Tom and Kai Gilb present a 4-day course to the whole R&D department and other related resources. By the end of November 2004, with the second release (9.0), we confirmed that the method can, consistently and repetitively, produce the results we need to lead our industry. We now expect the next versions of Confirmit will prove that we have matured in our understanding and execution of Evo.

REFERENCES

Gilb, Kai, *Evo: Evolutionary Project Management & Product Development, how successful people manage projects - delivering stakeholder values – requirements - project planning*. Book manuscript to be found at www.gilb.com

Gilb, Tom, *Competitive Engineering: A Handbook for Systems Engineering, Requirements Engineering, and Software Engineering using Planguage*. Elsevier Butterworth-Heinemann 2005, ISBN 0750665076.

Johnson, J., Keynote speech at XP 2002, Sardinia, Italy, 2002. Download talk via http://ciclamino.dibe.unige.it/xp2002/ [Last Accessed: April 2005].

Larman, Craig, *Agile and Iterative Development: A Manager's Guide*. Addison Wesley, 2003. Chapter 10 gives an excellent detailed description of the Evo method, and compares it to other agile methods like XP, Scrum and Crystal.

Taylor, A., "IT projects: sink or swim." *Computer Bulletin*, January 2000, British Computer Society. See http://www.bcs.org/BCS/Products/Publications/JournalsAndMagazines/ComputerBulletin/OnlineArchive/jan00/professionalpractice.htm [Last Accessed April 2005].

BIOGRAPHIES

Trond Johansen

Trond Johansen has an M.Sc. from the Norwegian Institute of Technology, University of Trondheim (NTNU). From 1998–1999, he worked as a system consultant for SAP at Hydro Data. He then worked from 1999-2000 as the product Manager for Nauticus CMC at Det Norske Veritas Software. In 2000, he joined Future Information Research Management (FIRM) as Quality Assurance & Process Manager, becoming Head of Project Management in 2005. His work has been related to process management and project management within software development, as well as to management in general.

URL: www.firmglobal.com Author Contact: trond.johansen@firmglobal.com, Oslo, Norway

Tom Gilb

Tom Gilb is the author of ‘Competitive Engineering: A Handbook for Systems & Software Engineering Management using Planguage’ (2005), ‘Principles of Software Engineering Management’ (1988) and ‘Software Inspection’ (1993). His book “Software Metrics” (1976) coined the term and, was used as the basis for the Software Engineering Institute Capability Maturity Model Level Four (SEI CMM Level 4). His most recent interests are development of true software engineering and systems engineering methods.

Tom Gilb was born in Pasadena CA in 1940. He moved to England in 1956, and then two years later he joined IBM in Norway. Since 1963, he has been an independent consultant and author. He is a member of INCOSE.

URL: www.gilb.com Author Contact: Tom@Gilb.com

The authors are extremely grateful for editing assistance on this paper from Lindsey Brodie, Middlesex University, UK.