Advanced Agile Practices - The Evo Method in Practice

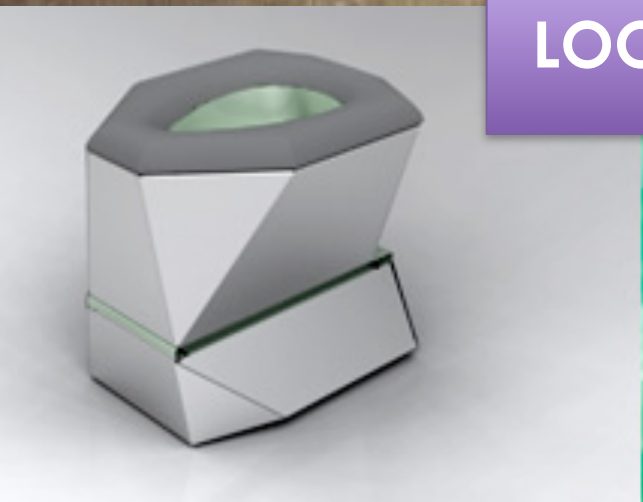
by Tom@Gilb.com
@ImTomGilb

MASTER 2016

The Agenda

- • The Evo Agile Startup Week: The US DoD Case
- • The Confirmit (Norway) Case Study: The Evo method in Practice
- • The Citigroup (London) Evo Project: Richard Smith
- This talk will give real case study insights into advanced successful delivery of quality and value.

LOOWATT: A NEW PARADIGM IN SANITATION



An Energy Producing Waterless Toilet System

Impact Estimation Table for Gates GCE Project

Key Values	Designs / Actions: Detailed risk assessment with associated	Research trip to madagascar (x3)	Detailed design research	Building financial models at	Research into existing	Creation of knowledge	Codification of our acquired knowledg	etc....	Total Impact	Safety Factor
	Impact (% progress towards target from given action)									
Improve Sanitation Target: 25% - 75% Unit: Waste collected / waste produced by user group	10	20	40	18	15	0	0		103	1.03
Sustainability and Longevity Target: 0$ - 0$ Unit: Cost to single user per month	0	5	20	50	10	0	0		85	0.85
Story and Data Target: 0.4 - 0.8 Unit: Average of factors rated 0.0 – 1.0	5	35	20	15	3	15	5		98	0.98
Managing Risk Target: 0.2 – 0.8 Unit: Average of factors rated 0.0 – 1.0	50	20	20	15	15	0	3		123	1.23
Methodology Target: 0.4 – 0.8 Unit: Average of factors rated 0.0 – 1.0	15	0	0	0	0	0	10		25	0.25
Diffusing Knowledge Target 0.15 – 0.8 Unit: Average of factors rated 0.0 – 1.0	0	8	0	0	10	50	15		83	0.83
Total impact of design / action	80	88	100	98	53	65	33	0		
Total cost of design / action (person days)	8	30	20	15	5	15	4	0		
Benefit to cost ratio	10	2.9	5.0	6.5	10.6	4.3	8.3	#####		

Key Values: LooWat

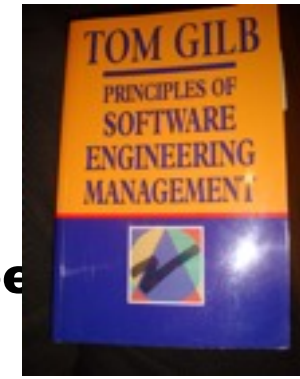
- **Improve Sanitation**
  Target: 25% - 75%
  Unit: Waste collected / waste produced by user group
- **Sustainability and Longevity**
  Target: 0$ - 0$
  Unit: Cost to single user per month
- **Story and Data**
  Target: 0.4 - 0.8
  Unit: Average of factors rated 0.0 – 1.0
- **Managing Risk**
  Target: 0.2 – 0.8
  Unit: Average of factors rated 0.0 – 1.0
- **Methodology**
  Target: 0.4 – 0.8
  Unit: Average of factors rated 0.0 – 1.0
- **Diffusing Knowledge**
  Target 0.15 – 0.8
  Unit: Average of factors rated 0.0 – 1.0

Winners!

- The **Bill & Melinda Gates Foundation** has awarded Loowatt Ltd a $1 million grant to expand its pioneering waterless toilet systems in Madagascar and Sub-Saharan Africa.
- 13.09.2013

Agile Credibility

- **Agile 'Grandfather' (Tom)**
  - **Practicing 'Agile' IT Projects since 1960**
  - **Preaching Agile since 1970's (CW UK)**
  - **Acknowledged Pioneer by Agile Gurus and Research**
    - **Beck, Sutherland, Highsmith, Cohn, Larman etc.**
    - **Ask me for details on this! I am too shy to show it here!**
- **Agile Practice**
  - **IT: for decades (Kai and Tom)**
  - **Organisations: for Decades (Citigroup, Intel, HP, Boe**
- **Books:**
  - **Principles of Software Engineering Management (1988) the book Beck and others refer to**
  - **Competitive Engineering (2005)**
  - **Evo: (Kai, evolving, 55 iterations)**



OK I am not that shy!

Agile References:

"Tom Gilb invented Evo, arguably the first Agile process. He and his son Kai have been working with me in Norway to align what they are doing with Scrum.

Kai has some excellent case studies where he has acted as Product Owner. He has done some of the most innovative things I have seen in the Scrum community."

Jeff Sutherland, co-inventor of Scrum, 5Feb 2010 in Scrum Alliance Email.

"Tom Gilb's Planguage referenced and praised at #scrumgathering by Jeff Sutherland. I highly agree" Mike Cohn, Tweet, Oct 19 2009

"I've always considered Tom to have been the original agilist. In 1989, he wrote about short iterations (each should be no more than 2% of the total project schedule). This was long before the rest of us had it figured out." Mike Cohn http://blog.mountaingoatsoftware.com/?p=77

Comment of Kent Beck on Tom Gilb's book , "Principles of Software Engineering Management": " A strong case for evolutionary delivery – small releases, constant refactoring, intense dialog with the customer". (Beck, page 173).

In a mail to Tom, Kent wrote: "I'm glad you and I have some alignment of ideas. I stole enough of yours that I'd be disappointed if we didn't :-), Kent" (2003)

Jim Highsmith (an Agile Manifesto signatory) commented: "Two individuals in particular pioneered the evolution of iterative development approached in the 1980's – Barry Boehm with his Spiral Model and Tom Gilb with his Evo model. I drew on Boehm's and Gilb's ideas for early inspiration in developing Adaptive Software Development. .... Gilb has long advocated this more explicit (quantitative) valuation in order to capture the early value and increase ROI" (Cutter It Journal: The Journal of Information Technology Management, July 2004page 4, July 2004).

111111

The Unity Method 111111

for decomposition into iterative value delivery steps

By Tom@Gilb.com

Slides at www.gilb.com/downloads

http://www.gilb.com/tiki-download_file.php?fileId=451

Originally made as

10 minute lightening talk

www.smidig.no

´1´ 4 U2

One Bono U2

´One´ lyrics

Is it getting better?

Or do you feel the same?

Will it make it easier on you now?

You got someone to blame

You say, one love, one life

When it's one need in the night

One love, we get to share it

Leaves you baby if you don't care for it

One love, one blood

One life, you got to do what you should

One life, with each other

Sisters, brothers

One life but we're not the same

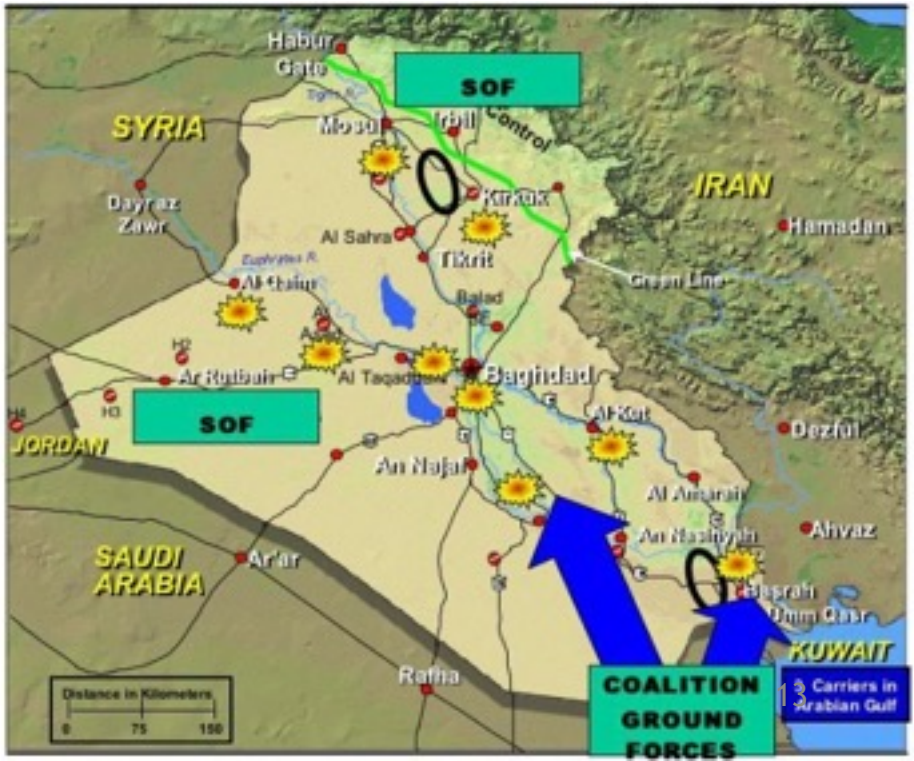
We get to carry each other, carry each other

One

One

A True War Story
111111 in practice

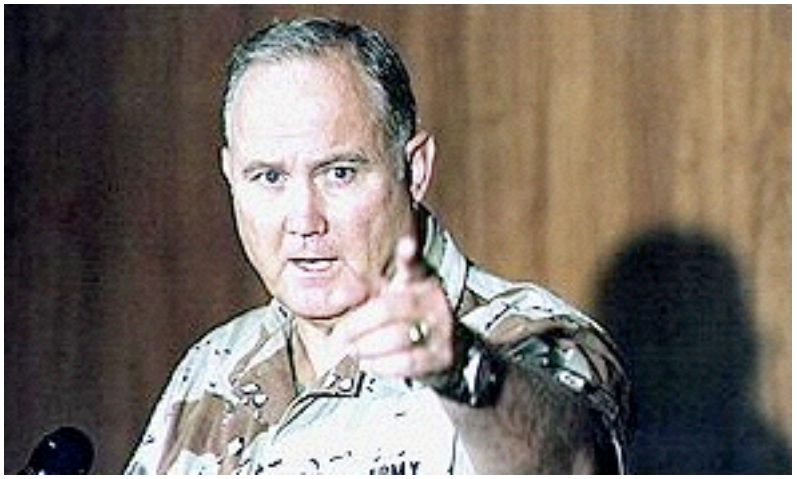
- How we found a value delivery step ´next week´
  - a week of value delivery beat 11 years of waterfall method



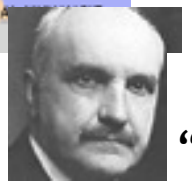
The *Persinscom IT System* Case

Commanding General
Norman Schwartzkopf

´Stormin´ Norman´



He who does not learn from history
Is doomed to repeat it

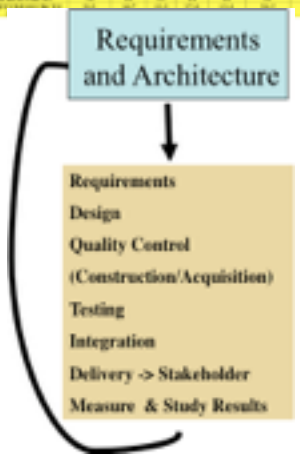


A Man Who understood that
"a bird in the hand is worth two in the Bush" <-tsg

The ´Evo´ Planning Week at DoD

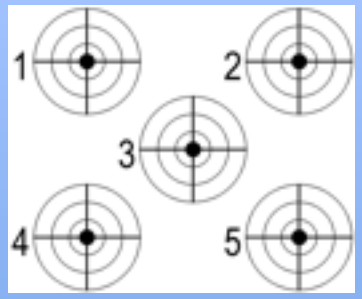
- Monday
  - Define top Ten critical objectives, quantitatively
  - Agree that thee are the main points of the effort/project
- Tuesday
  - Define roughly the top ten most powerful strategies
  - for enabling us to reach our objectives on time
- Wednesday
  - Make an Impact Estimation Table for Objectives/Strategies
  - Sanity Test: do we seem to have enough powerful strategies to get to our Goals, with a reasonable safety margin?
  - A tool for decomposing the value steps and seeing best value for resources
- Thursday
  - Divide into rough delivery steps (annual, quarterly)
  - Derive a delivery step for 'Next Week'
- Friday
  - Present these plans to approval manager (Brigadier General Pellicci)
  - get approval to deliver next week
  - (they can´t resist results next week!

US Army Example: PERSINSCOM



US Army Example: PERSINSCOM: Personnel System

STRATEGIES ➔ OBJECTIVES
Customer Service ?➔0 Violation of agreement
Availability 90% ➔ 99.5% Up time
Usability 200 ➔ 60 Requests by Users
Responsiveness 70% ➔ ECP's on time
Productivity 3:1 Return on Investment
Morale 72 ➔ 60 per mo. Sick Leave
Data Integrity 88% ➔ 97% Data Error %
Technology Adaptability 75% Adapt Technology
Requirement Adaptability ? ➔ 2.6% Adapt to Change
Resource Adaptability 2.1M ➔ ? Resource Change
Cost Reduction FADS ➔ 30% Total Funding



Monday
←The Top Ten
Critical
Objectives
Were decided

Sample of Objectives/Strategy definitions
US Army Example: PERSINSCOM: Personnel System

- *Example of one of the Objectives:*

Customer Service:

Type: Critical Top level Systems Objective

Gist: Improve customer perception of quality of service provided.

Scale: Violations of Customer Agreement per Month.

Meter: Log of Violations.

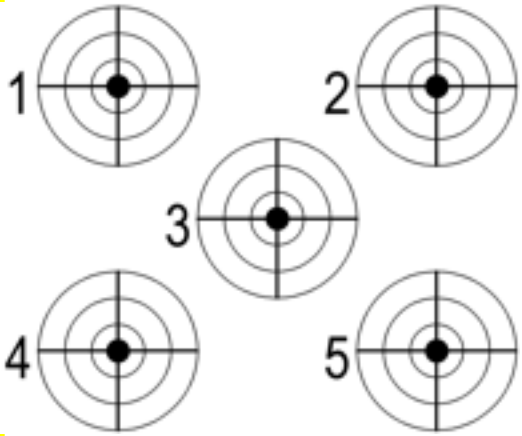
Past [Last Year] Unknown Number ←State of PERSCOM Management Review

Record [NARDAC] 0 ? ← NARDAC Reports Last Year

Fail : <must be better than Past, Unknown number> ←CG

Goal [This Year, PERSINCOM] 0 "Go for the Record" ← Group SWAG

.



US Army Example: PERSINSCOM: Personnel System

STRATEGIES ➔ OBJECTIVES	Technology Investment	Business Practices	People	Empow-erment	*Principles of IMA Management*	Business Process Re-engineering	SUM
Customer Service ?➔0 Violation of agreement							
Availability 90% ➔ 99.5% Up time							
Usability 200 ➔ 60 Requests by Users							
Responsiveness 70% ➔ ECP's on time							
Productivity 3:1 Return on Investment							
Morale 72 ➔ 60 per mo. Sick Leave							
Data Integrity 88% ➔ 97% Data Error %							
Technology Adaptability 75% Adapt Technology							
Requirement Adaptability ? ➔ 2.6% Adapt to Change							
Resource Adaptability 2.1M ➔ ? Resource Change							
Cost Reduction FADS ➔ 30% Total Funding							

Tuesday
The Top Ten
Critical **Strategies**
For reaching the
←objectives
Were decided

A Strategy (Top Level of Detail)

Technology Investment:

Gist: Exploit investment in high return technology.

Impacts: productivity, customer service and conserves resources.

Wednesday: Sanity Check
Day 3 of 5 of 'Feasibility Study

- **We made a rough evaluation**
  - **of how powerful our strategies might be**
  - **in relation to our objectives**

- **Impact Estimation Table**
  - **0% Neutral, no ± impact**
  - **100% Gets us to Goal level on time**
  - **50% Gets us half way to Goal at deadline**
  - **-10% has 10% negative side effect**

STRATEGIES → OBJECTIVES	Technology Investment	Business Practices	People	Empow-erment	*Principles of IMA Management*	Business Process Re-engineering	SUM
Customer Service ?→0 Violation of agreement	50%	10%	5%	5%	5%	60%	185%
Availability 90% → 99.5% Up time	50%	5%	5-10%	0	0	200%	265%
Usability 200 → 60 Requests by Users	50%	5-10%	5-10%	50%	0	10%	130%
Responsiveness 70% → ECP's on time	50%	10%	90%	25%	5%	50%	180%
Productivity 3:1 Return on Investment	45%	60%	10%	35%	100%	53%	303%
Morale 72 → 60 per mo. Sick Leave	50%	5%	75%	45%	15%	61%	251%
Data Integrity 88% → 97% Data Error %	42%	10%	25%	5%	70%	25%	177%
Technology Adaptability 75% Adapt Technology	5%	30%	5%	60%	0	60%	160%
Requirement Adaptability ? → 2.6% Adapt to Change	80%	20%	60%	75%	20%	5%	260%
Resource Adaptability 2.1M → ? Resource Change	10%	80%	5%	50%	50%	75%	270%
Cost Reduction FADS → 30% Total Funding	50%	40%	10%	40%	50%	50%	240%
SUM IMPACT FOR EACH SOLUTION	***482%***	***280%***	***305%***	***390%***	***315%***	***649%***	
Money % of total budget	15%	4%	3%	4%	6%	4%	
Time % total work months/year	15%	15%	20%	10%	20%	18%	
SUM RESOURCES	***30***	***19***	***23***	***14***	***26***	***22***	
BENEFIT/RESOURCES RATIO	***16:1***	**14:7**	**13:3**	**27:9**	**12:1**	**29:5**	

MEASURING HAND FOR GLOVE SIZE

US DoD. Persinscom **Impact EstimationTable**:

Designs

Requirements

Design Ideas ->	*Technology Investment*	*Business Practices*	*People*	*Empowerment*	*Principles of IMA Management*	*Business Process Re-engineering*	*Sum Requirements*
	50%	1[illegible]	5%	5%	5%	60%	185%
Availability 90% <-> 99.5% Up time	50%		5–10%	0%	0%	200%	265%
Usability 200 <-> 60 Requests by Users			5–10%	50%	0%	10%	130%
Responsiveness 70% <-> ECP's on time	50%	10%	90%	25%	5%	50%	180%
Productivity 3:1 Return on Investment	45% 50%						303% 251%
Morale 72 <-> 60 per month on Sick Leave							
Data Integrity 88% <-> 97% Data Error %	42%						177%
Technology Adaptability 75% Adapt Technology	5%						160%
Requirement Adaptability ? <-> 2.6% Adapt to Change	80%						260%
Resource Adaptability 2.1M <-> ? Resource Change	10%	80%	5%	50%	50%	75%	270%
Cost Reduction FADS <-> 30% Total Funding	50%	40%	10%	40%	50%	50%	240%
Sum of Performance	*482%*	*280%*	*305%*	*390%*	*315%*	*649%*	
Money % of total budget	15%	4%	3%	4%	6%	4%	36%
Time % total work months/year	15%	15%	20%	10%	20%	18%	98%
Sum of Costs	*30*	*19*	*23*	*14*	*26*	*22*	
Performance to Cost Ratio	*16:1*	*14:7*	*13:3*	*27:9*	*12:1*	29.5 :1	

Estimated Impact of

Design
-> Requirements

US Army Example: PERSINSCOM: Personnel System

STRATEGIES → **OBJECTIVES**	Technology Investment	Business Practices	People	Empow-erment	*Principles of IMA Management*	Business Process Re-engineering	SUM
Customer Service ?→0 Violation of agreement	50%	10%	5%	5%	5%	60%	185%
Availability 90% → 99.5% Up time	50%	5%	5-10%	0	0	200%	265%
Usability 200 → 60 Requests by Users	50%	5-10%	5-10%	50%	0	10%	130%
Responsiveness 70% → ECP's on time	50%	10%	90%	25%	5%	50%	180%
Productivity 3:1 Return on Investment	45%	60%	10%	35%	100%	53%	303%
Morale 72 → 60 per mo. Sick Leave	50%	5%	75%	45%	15%	61%	251%
Data Integrity 88% → 97% Data Error %	42%	10%	25%	5%	70%	25%	177%
Technology Adaptability 75% Adapt Technology	5%	30%	5%	60%	0	60%	160%
Requirement Adaptability ? → 2.6% Adapt to Change	80%	20%	60%	75%	20%	5%	260%
Resource Adaptability 2.1M → ? Resource Change	10%	80%	5%	50%	50%	75%	270%
Cost Reduction FADS → 30% Total Funding	50%	40%	10%	40%	50%	50%	240%
SUM IMPACT FOR EACH SOLUTION	***482%***	***280%***	***305%***	***390%***	***315%***	***649%***	
Money % of total budget	15%	4%	3%	4%	6%	4%	
Time % total work months/year	15%	15%	20%	10%	20%	18%	
SUM RESOURCES	***30***	***19***	***23***	***14***	***26***	***22***	
BENEFIT/RESOURCES RATIO	***16:1***	**14:7**	**13:3**	**27:9**	**12:1**	29.5 :1	

Impact Estimation: Value-for-Money Delivery Table

STRATEGIES ➔ OBJECTIVES	Technology Investment	Business Practices	People	Empow-erment	*Principles of IMA Management*	Business Process Re-engineering	SUM
Customer Service ?➔0 Violation of agreement	50%	10%	5%	5%	5%	60%	185%
Availability 90% ➔ 99.5% Up time	50%	5%	5-10%	0	0	200%	265%
Usability 200 ➔ 60 Requests by Users	50%	5-10%	5-10%	50%	0	10%	130%
Responsiveness 70% ➔ ECP's on time	50%	10%	90%	25%	5%	50%	180%
Productivity 3:1 Return on Investment	45%	60%	10%	35%	100%	53%	303%
Morale 72 ➔ 60 per mo. Sick Leave	50%	5%	75%	45%	15%	61%	251%
Data Integrity 88% ➔ 97% Data Error %	42%	10%	25%	5%	70%	25%	177%
Technology Adaptability 75% Adapt Technology	5%	30%	5%	60%	0	60%	160%
Requirement Adaptability ? ➔ 2.6% Adapt to Change	80%	20%	60%	75%	20%	5%	260%
Resource Adaptability 2.1M ➔ ? Resource Change	10%	80%	5%	50%	50%	75%	270%
Cost Reduction FADS ➔ 30% Total Funding	50%	40%	10%	40%	50%	50%	240%
SUM IMPACT FOR EACH SOLUTION	***482%***	***280%***	***305%***	***390%***	***315%***	***649%***	
Money % of total budget	15%	4%	3%	4%	6%	4%	
Time % total work months/year	15%	15%	20%	10%	20%	18%	
SUM RESOURCES	***30***	***19***	***23***	***14***	***26***	***22***	
BENEFIT/RESOURCES RATIO	***16:1***	**14:7**	**13:3**	**27:9**	**12:1**	**29.5 : 1**	

Thursday:
Day 4 of 5 of 'Feasibility Study

- We looked for a way to deliver some stakeholder results, next week
- **1 1 1 1 1 1 1 Unity**
  - **1% increase at least**
  - **1 stakeholder**
  - **1 quality/value**
  - **1 week delivery cycle**
  - **1 function focus**
  - **1 design used**

STRATEGIES → **OBJECTIVES**	Technology Investment	Business Practices	People	Empow-erment	*Principles of IMA Management*	Business Process Re-engineering	SUM
Customer Service ?→0 Violation of agreement	50%	10%	5%	5%	5%	60%	185%
Availability 90% → 99.5% Up time	50%	5%	5-10%	0	0	200%	265%
Usability 200 → 60 Requests by Users	50%	5-10%	5-10%	50%	0	10%	130%
Responsiveness 70% → ECP's on time	50%	10%	90%	25%	5%	50%	180%
Productivity 3:1 Return on Investment	45%	60%	10%	35%	100%	53%	303%
Morale 72 → 60 per mo. Sick Leave	50%	5%	75%	45%	15%	61%	251%
Data Integrity 88% → 97% Data Error %	42%	10%	25%	5%	70%	25%	177%
Technology Adaptability 75% Adapt Technology	5%	30%	5%	60%	0	60%	160%
Requirement Adaptability ? → 2.6% Adapt to Change	80%	20%	60%	75%	20%	5%	260%
Resource Adaptability 2.1M → ? Resource Change	10%	80%	5%	50%	50%	75%	270%
Cost Reduction FADS → 30% Total Funding	50%	40%	10%	40%	50%	50%	240%
SUM IMPACT FOR EACH SOLUTION	***482%***	***280%***	***305%***	***390%***	***315%***	***649%***	
Money % of total budget	15%	4%	3%	4%	6%	4%	
Time % total work months/year	15%	15%	20%	10%	20%	18%	
SUM RESOURCES	***30***	***19***	***23***	***14***	***26***	***22***	
BENEFIT/RESOURCES RATIO	***16:1***	**14:7**	**13:3**	**27:9**	**12:1**	**29:5**	

Next weeks Evo Step??

- **“You won’t believe we never thought of this, Tom!’**

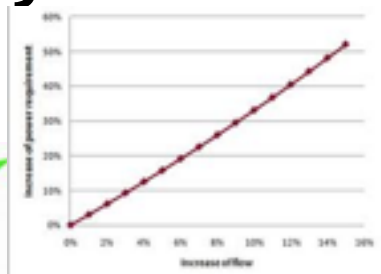
- **The step:**
  - **When the Top General Signs in**
  - **Move him to the head of the queue**
    - **Of all people inquiring on the system.**

- **Can you deliver it next week?**
  - **Its already done: If General, move to head of queue’**

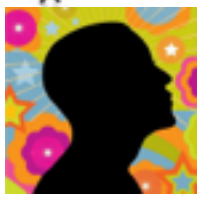
1 1 1 1 1 1 Unity

–1% increase at least

–1 stakeholder

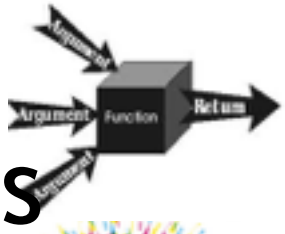


–1 quality or value

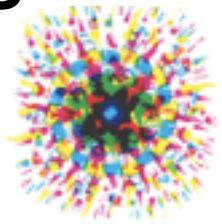


–1-week delivery cycle

–1 function focus



–1 design used



“I kill men for a living! (General Pellicci)

UNITED STATES ARMY PERSONNEL INFORMATION SYSTEMS COMMAND

CERTIFICATE of APPRECIATION

is awarded to

MR. TOM GILB

for

SELFLESS AND DEDICATED SERVICE IN SUPPORT OF THE PERSONNEL INFORMATION SYSTEMS COMMAND. AS A MANAGEMENT CONSULTANT IN RESULT DELIVERY PLANNING, HIS PATRIOTISM, PROFESSIONAL COMPETENCE AND PERSONAL SACRIFICES ARE HIGHLY COMMENDABLE. TOM GILB'S DEDICATION AND THE EXCEPTIONAL MANNER IN WHICH HE PERFORMED HIS DUTIES HAD A DIRECT AND SIGNIFICANT IMPACT ON PERSINSCOM'S MISSION. HIS OUTSTANDING CONTRIBUTIONS AND DISTINGUISHED SERVICE REFLECT GREAT CREDIT ON HIM AND THE UNITED STATES ARMY. CONGRATULATIONS FOR A JOB WELL DONE.

30 AUGUST 1991

Personnel Information Systems Command

Jack A. Pellicci

JACK A. PELLICCI
Brigadier General, USA
Commanding

The Evo Startup Process a practical example

- The ‘standards for Startup are at
  - Evo Startup Standard, Jan 12 2013
  - http://www.gilb.com/dl562
-
- Evo Project Management Standard, Jan 12 2013
- http://www.gilb.com/dl563

Startup Process Day 1 and 2

- Day 1: **Project Objectives**: The top few critical objectives quantified.
  - Objective: Determine, clarify, agree critical few project objectives – results – end states
  - **Process**:
    - Analyze current documentation and slides, for expressed or implied objectives (often implied by designs or lower level objectives)
    - Develop list of **Stakeholders** and their needs and values
    - Brainstorm 'top ten' critical objectives names list. Agree they are top critical few.
    - Detail definition in Planguage – meaning quantify and define clearly, unambiguously and in detail (a page)
    - Quality Control Objectives for Clarity: Major defect measurement. Exit if less than 1.0 majors per page
    - Quality Control Objectives for Relevance: Review against higher level objectives than project for alignment.
    - Define Constraints: resources, traditions, policies, corporate IT architecture, hidden assumptions.
    - Define Issues – yet unresolved
    - Note we might well choose to several things in *parallel*.
  - **Output**: A solid set of the top few critical *objectives* in quantified and measurable language. *Stakeholder* data specified.
  - **Participants**: anybody who is concerned with the business results, the higher the management level the better.
  - **End of Day Process**: meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.
  - **Note**: this process is so critical and can be time consuming, so if necessary it can spill over to next day. Perhaps in parallel with startup of the strategy identification. *Nothing is more critical or fundamental than doing this well.*
- Day 2: **Project Strategies and Architecture**: the top few critical strategies for reaching the critical objectives
  - **Objective**: to identify the top 'ten' most critical strategic decisions or architectures; the ones that will contribute or enable us most, to reach our primary objective goal levels on time.
  - **Process**:
    - Analysis of current documentation and slides to identify candidate strategies, implied or expressed.
    - Brainstorming of the 'names' of the specific strategy list, the top ten and a set of less powerful ideas (say 11-30)
    - Detail each top ten strategy sufficiently to understand impacts (on objectives, time and costs)
    - Specify, for each strategy all critical related information (like stakeholders, risks, assumptions, constraints, etc.)
    - Quality Control for clarity – correct unclear items. Exit based on defect level, or not.
    - Likely that work will need to be done in parallel in order to do ten strategies to a rich level of specification.
  - **Output**: A formal strategy specification, ready for evaluation, and decomposition and delivery of partial value results.
  - **Participants**: system architects, project architects, strategy planners. And members of the project team who will be in on the entire weeks process. The major input here is technical and organizational strategy (the means to reach the objectives)
  - **End of Day Process: :** meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.

Startup Process
Day 3 and 4

Day 3: Evaluation of Strategies using Impact Estimation: our best estimates with experience and risk. How sure are of the major strategy decisions.

- **Objective**: to estimate to primary effects and all side effects of all top critical strategies on all top critical objectives, and on some resources (time, cost, effort). The estimates will be backed up by evidence, or their credibility will be rated low.
- **Process**:
  - Using the objectives and strategies developed on first 2 days as inputs
  - Populate an Impact Estimation table (aka Value Decision Table) with estimates of the expected result of deploying defined strategies. Estimate main intended impacts
  - And all side effects (on other core objectives)
  - And on all resources (time, money. Effort)
  - Estimate ± ranges
  - Specify evidence and sources for estimates
  - Determine Credibility level
  - Quality Control the IE table against standards (Rules for IE in CE book), for possible 'exit' (meets standards)
  - Lots of parallel work needed and expected to do a good job.
- **Output**:
  - A fairly decent Impact Estimation table, possibly a several level set of them.
    - This will tell us if it is safe to proceed (we have good enough strategies)
    - And it will help us prioritize high value deliveries soon.
- **Participants**: architects, planners, anybody with strong views on any of the strategies. The team for the week.
- **Note**: *it might be necessary and desirable, now or later, to do this impact estimation process at 2 or 3 related levels (Business, Stakeholder, IT System) in order to see the Business-IT relationship clearly. This might exceed time limits and be done parallel or later.*
- **End of Day Process**: meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.

Day 4: **Evolutionary Step Decomposition**: what are the high value short term value delivery steps we can execute.

- **Objective**: to identify near team candidates for real value delivery to real stakeholders. What can we do for real next week!
- **Process**:
  - Identify highest value (to costs) strategies and sub-sets of strategies
  - Decompose into doable subsets in weekly to monthly cycles of result delivery
  - Plan the near steps (1 or more) in detail so that we are ready to execute the step in practice.
    - Who does it, main responsible, team.
    - Expected measurable results and costs
    - Stakeholder involved in receiving
    - Test process (for value)
- **Output**: 1 or more potential steps for value delivery to some stakeholders, a plan good enough to approve and execute in practive.
- **Participants**: Project Management, architects prepared to decompose architecture in practice. The weeks team for this start up study.
- **End of Day Process:** meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.

Day 5

- Boss approves doing the next week

The ‘Evo’ (**Evo**lutionary) Method for Project Management.

The ‘Evo’ (Evolutionary) Method for Project Management.

Process Description , http://www.gilb.com/dl563

1. Gather from all the key stakeholders the top few (5 to 20) most critical goals that the project needs to deliver.

Give each goal a reference name (a tag).

2. For each goal, define a scale of measure and a ‘final’ goal level.

For example: Reliable: Scale: Mean Time Before Failure, Goal: 1 month.

3. Define approximately 4 budgets for your most limited resources

(for example, time, people, money, and equipment).

4. Write up these plans for the goals and budgets

(Try to ensure this is kept to only one page).

5. Negotiate with the key stakeholders to formally agree the goals and budgets.

6. Plan to deliver some benefit

(that is, progress towards the goals)

in weekly (or shorter) increments (Evo steps).

7. Implement the project in Evo steps.

Report to project sponsors after each Evo step (weekly, or shorter) with your best available estimates or measures, for each performance goal and each resource budget.

On a single page, summarize the progress to date towards achieving the goals and the costs incurred.

8. When all Goals are reached: ‘Claim success and move on’

a. Free remaining resources for more profitable ventures

The Confirmit Case Study 2003-2013
Agile Quantified Value Delivery

See paper on this case at www.gilb.com
Papers/Cases/Slides, Gilb Library,

value slide w… http://www.gilb.com/tiki-download_file.php?fileId=152
ppr wrong ag… http://www.gilb.com/tiki-download_file.php?fileId=50
Paper Firm http://www.gilb.com/tiki-download_file.php?fileId=32
And see papers (IEEE Software Fall 2006) by Geir K Hanssen, SINTEF

Their product = confirmit

Chief Storyteller = Trond Johansen

Here are some of the Clients of the Confirmit Product in 2003heading

We gave them a 1 day briefing on our Evo method and Planguage

That's all they needed to succeed!

They were Real engineers

Shift: from
‘Function’ to ‘Stakeholder Quality’
(They never went back to the burn down stack)

- **“Our new focus is on the day-to-day operations of our Market Research users,**
  - **not a list of features that they might or might not like. 50%are never used!**
  - **We KNOW that increased efficiency, which leads to more profit, will please them.**
  - **The ‘45 minutes actually saved x thousands of customer reports’**
    - **= big $$$ saved**
- **After one week we had defined more or less all the requirements for t t version (8.5) of Confirmit. “**
  - **Trond Johansen**

Trond Johansen

EVO Plan Confirmit 8.5 in **Evo Step Impact *Measurement***

4 product areas were attacked in all: **25 Qualities** concurrently, one quarter of a year. Total development staff = 13

9

8

3

3

Impact Estimation Table: Reportal codename "Hyggen"

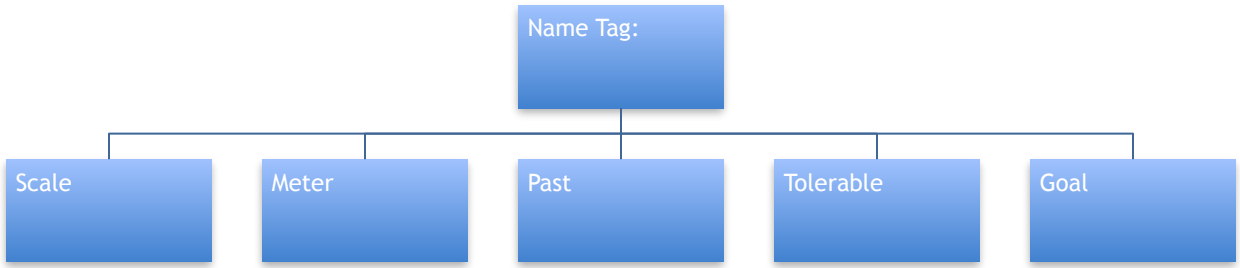
Current Status	Improvements		Reportal - E-SAT features		
Units	Units	%	Past	Tolerable	Goal
			Usability.Intuitivness (%)		
75,0	25,0	62,5	50	75	90
			Usability.Consistency.Visual (Elements)		
14,0	14,0	100,0	0	11	14
			Usability.Consistency.Interaction (Components)		
15,0	15,0	107,1	0	11	14
			Usability.Productivity (minutes)		
5,0	75,0	96,2	80	5	2
5,0	45,0	95,7	50	5	1
			Usability.Flexibility.OfflineReport.ExportFormats		
3,0	2,0	66,7	1	3	4
			Usability.Robustness (errors)		
1,0	22,0	95,7	7	1	0
			Usability.Replacability (nr of features)		
4,0	5,0	100,0	8	5	3
			Usability.ResponseTime.ExportRep (min		es
1,0	12,0	150,0	13	13	5
			Usability.ResponseTime.ViewRepo seco		s)
1,0	14,0	100,0	15		1
			Development resources		
203,0			0		91

Current Status	Improvements		Survey Engine .NET		
Units	Units	%	Past	Tolerable	Goal
			Backwards.Compatibility (%)		
83,0	48,0	80,0	40	85	95
0,0	67,0	100,0	67	0	0
			Generate.WI.Time (small/medium/large seconds)		
4,0	59,0	100,0	63	8	4
10,0	397,0	100,0	407	100	10
94,0	2290,0	103,9	2384	500	180
			Testability (%)		
10,0	10,0	13,3	0	100	100
			Usability.Speed (seconds/user rating 1-10)		
774,0	507,0	51,7	1281	600	300
5,0	3,0	60,0	2	5	7
			Runtime.ResourceUsage.Memory		
0,0	0,0	0,0		?	?
			Runtime.ResourceUsage.CPU		
3,0	35	97,2	38	3	2
			Runtime.ResourceUsage.MemoryLeak		
0	800	100,0	800	0	0
			Runtime.Concurrency (number of users)		
350	1100	146,7	150	500	1000
			Development resources		
64			0		84

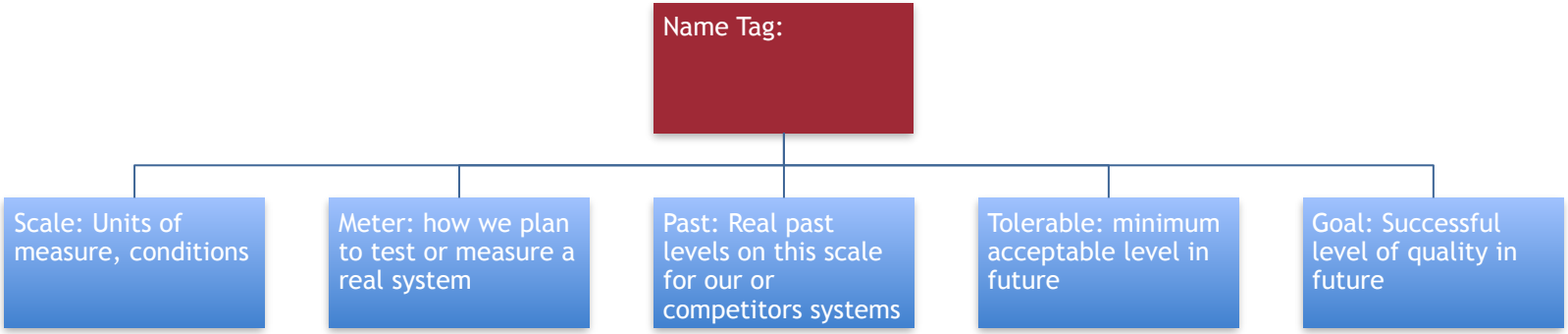
Current Status	Improvements		Reportal - MR Features		
Units	Units	%	Past	Tolerable	Goal
			Usability.Replacability (feature count)		
1,0	1,0	50,0	14	13	12
			Usability.Productivity (minutes)		
20,0	45,0	112,5	65	35	25
			Usability.ClientAcceptance (features count)		
4,4	4,4	36,7	0	4	12
			Development resources		
101,0			0		86

Current Status	Improvements		XML Web Services		
Units	Units	%	Past	Tolerable	Goal
			TransferDefinition.Usability.Efficiency		
7,0	9,0	81,8	16	10	5
17,0	8,0	53,3	25	15	10
			TransferDefinition.Usability.Response		
943,0	-186,0	######	170	60	30
			TransferDefinition.Usability.Intuitiveness		
5,0	10,0	95,2	15	7,5	4,5
			Development resources		
2,0			0		48

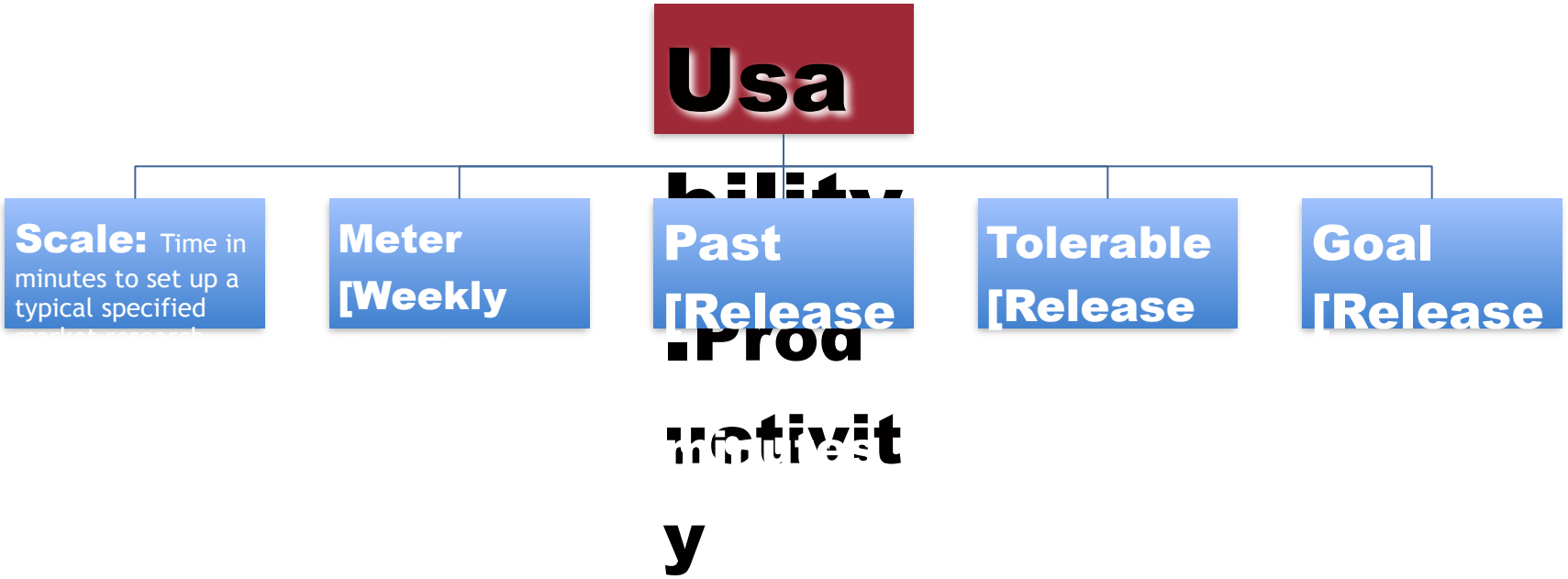
Each of the 25 Initial Quality Requirement has this ‘Planguage’ format



Each Quality Requirement has this 'Planguage' format: Meaning



Each Quality Requirement has this 'Planguage' format: Real Example



Real Example of 1 of the 25 Quality Requirements

Usability.Productivity (taken from Confirmit 8.5, performed a set of predefined steps, to produce a standard MR Report.

development)

Scale for quantification: Time in minutes to set up a typical specified Market Research-report

Past Level [Release 8.0]: 65 mins.,

Tolerable Limit [Release 8.5]: 35 mins.,

Goal [Release 8.5]: 25 mins.

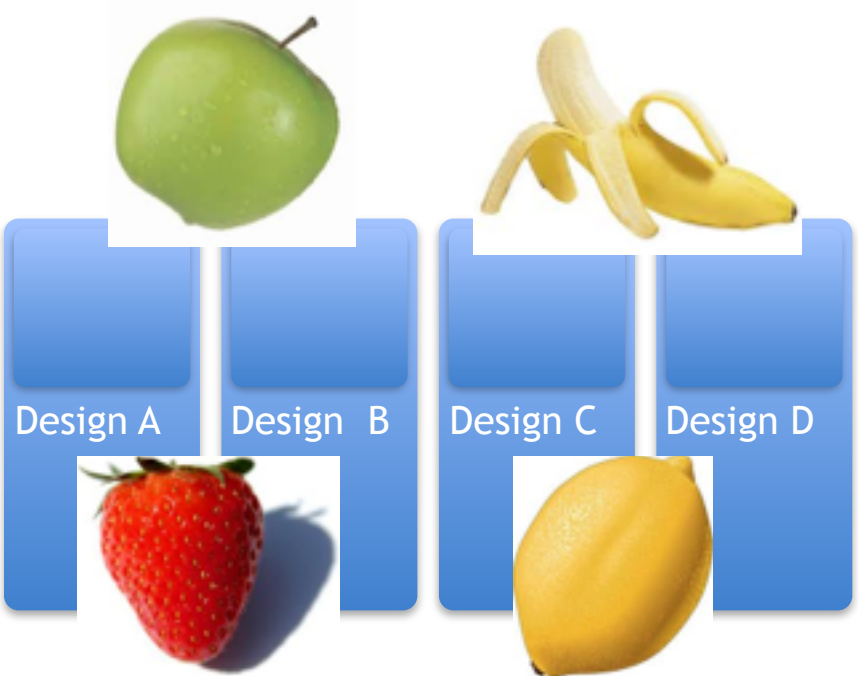
Note: end result was actually 20 minutes ☺

Meter [Weekly Step]: Candidates with Reportal experience, and with knowledge of MR-specific reporting features

Trond Johansen

Design Process

Design Suggestions

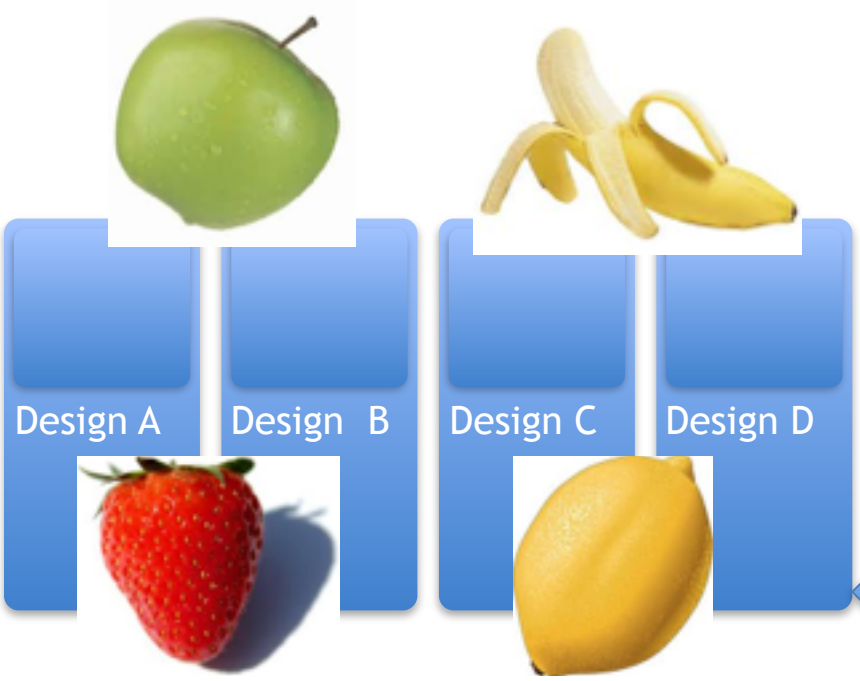


Impacts to Cost Evaluation

	A	B	C	D
Goal 1	30%	10%’	-10%	80%
Cost	10	50	1	20
G/C	3:1	1:5	?	4:1

Design Process: The winner

Design Suggestions



Impacts to Cost Evaluation

	A	B	C	D
Goal 1	30%	10%’	-10%	80%
Cost	10	50	1	20
G/C	3:1	1:5	?	4:1

Trond Johansen

- **IET for Market Research Product – Confirmit 8.5**
- **Solution: 'Recoding' (of MR codes)**
  - **Make it possible to recode variables on the fly from Reportal.**
  - **Estimated effort: 4 days**
  - **<u>Estimated</u> Productivity Improvement: 20 minutes (50% way to Goal)**
  - **actual result 38 minutes (95% progress towards Goal)**

	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

Quantified Value Delivery Project Management in a Nutshell

Quantified Value Requirements, Design, Design Value/cost estimation, Measurement of Value Delivery, Incremental Project Progress to Date

	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	[illegible]	%	[illegible]	
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

Weekly Testing

Estimates

Priority Next week Warning metrics based

Cumulative weekly progress metric

Constraint

Target

Requirements

Past	Tolerable	Goal
Usability.Replacability (feature count)		
2	1	0
Usability.Speed.NewFeaturesImpact (%)		
0	15	5
0	15	5
0	30	10
Usability.Intuitiveness (%)		
0	60	80
Usability.Productivity (minutes)		
65	35	25
Development resources		
		110

Benchmark

Constraint

Target

Cycle Resource

Design Engineering

We estimate the 'design effect' at beginning of week

And measure the actual effect, at the end of the week

Step9			
Recoding			
Estimated impact		Actual impact	
Units	%	Units	%
20,00	50,00	38,00	95,00
Minutes	% way to Goal	Minutes	% way to Goal
4,00	3,64	4,00	3,64

Estimates

Meter/Test

Weekly

Work days

% of Time to Release

Week 9 of 12 Before Release

Tag of a 'designi dea'

Tracking Progress: after each Evo value delivery cycle

	Current Status	Improvements	
	Units	Units	%
	1,00	1,0	50,0
	5,00	5,0	100,0
	10,00	10,0	200,0
	0,00	0,0	0,0
	0,00	0,0	0,0
	20,00	45,0	112,5
		101,0	91,8

<- 50% of way to Goal level

<- All the way to the goal

<- Twice the way to the Goal level

<- No progress from Past level

<- 12.5 % over

Computing Current Priority for next resources.

'Dynamic Prioritization'

	Current Status	Improvements	
	Units	Units	%
	1,00	1,0	50,0
	5,00	5,0	100,0
	10,00	10,0	200,0
	0,00	0,0	0,0
	0,00	0,0	0,0
	20,00	45,0	112,5
		101,0	91,8

Tolerable but not at Goal level

Not even Tolerable level Give this highest priority

No priority. You reached or exceeded Goal

Overview of Evo Project Management using 'Impact Estimation' table

	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		%		
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)						
19		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

Weekly Testing

Estimates

Priority

Next week

Warning metrics based

Cumulative weekly progress metric

Constraint

Target

Concurrent Quantified ‘Empowered Creativity’ *

* Empowered Creativity: Term coined by Trond Johansen, Confirmit, 2003

EVO Plan Confirmit 8.5 in **Evo Step Impact *Measurement***

4 product areas were attacked in all: **25 Qualities** concurrently, one quarter of a year. Total development staff = 13

9

8

3

3

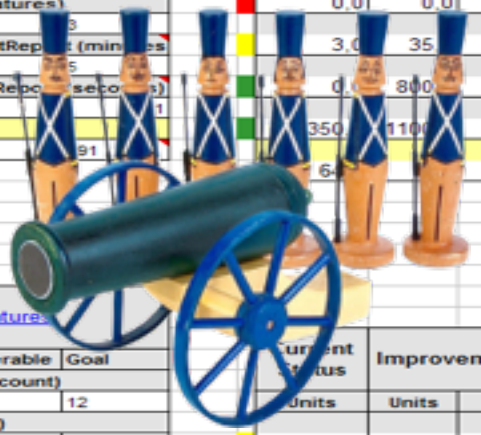
Impact Estimation Table: Reportal codename "Hyggen"

Current Status	Improvements		Reportal - E-SAT features		
Units	Units	%	Past	Tolerable	Goal
			Usability.Intuitivness (%)		
75,0	25,0	62,5	50	75	90
			Usability.Consistency.Visual (Elements)		
14,0	14,0	100,0	0	11	14
			Usability.Consistency.Interaction (Components)		
15,0	15,0	107,1	0	11	14
			Usability.Productivity (minutes)		
5,0	75,0	96,2	80	5	2
5,0	45,0	95,7	50	5	1
			Usability.Flexibility.OfflineReport.ExportFormats		
3,0	2,0	66,7	1	3	4
			Usability.Robustness (errors)		
1,0	22,0	95,7	7	1	0
			Usability.Replacability (nr of features)		
4,0	5,0	100,0	8	5	3
			Usability.ResponseTime.ExportRep (min		es
1,0	12,0	150,0	13	13	5
			Usability.ResponseTime.ViewRepo seco		s)
1,0	14,0	100,0	15		1
			Development resources		
203,0			0		91

Current Status	Improvements		Survey Engine .NET		
Units	Units	%	Past	Tolerable	Goal
			Backwards.Compatibility (%)		
83,0	48,0	80,0	40	85	95
0,0	67,0	100,0	67	0	0
			Generate.WI.Time (small/medium/large seconds)		
4,0	59,0	100,0	63	8	4
10,0	397,0	100,0	407	100	10
94,0	2290,0	103,9	2384	500	180
			Testability (%)		
10,0	10,0	13,3	0	100	100
			Usability.Speed (seconds/user rating 1-10)		
774,0	507,0	51,7	1281	600	300
5,0	3,0	60,0	2	5	7
			Runtime.ResourceUsage.Memory		
0,0	0,0	0,0		?	?
			Runtime.ResourceUsage.CPU		
3,0	35	97,2	38	3	2
			Runtime.ResourceUsage.MemoryLeak		
0	800	100,0	800	0	0
			Runtime.Concurrency (number of users)		
350	1100	146,7	150	500	1000
			Development resources		
64			0		84

Current Status	Improvements		Reportal - MR Features		
Units	Units	%	Past	Tolerable	Goal
			Usability.Replacability (feature count)		
1,0	1,0	50,0	14	13	12
			Usability.Productivity (minutes)		
20,0	45,0	112,5	65	35	25
			Usability.ClientAcceptance (features count)		
4,4	4,4	36,7	0	4	12
			Development resources		
101,0			0		86

Current Status	Improvements		XML Web Services		
Units	Units	%	Past	Tolerable	Goal
			TransferDefinition.Usability.Efficiency		
7,0	9,0	81,8	16	10	5
17,0	8,0	53,3	25	15	10
			TransferDefinition.Usability.Response		
943,0	-186,0	######	170	60	30
			TransferDefinition.Usability.Intuitiveness		
5,0	10,0	95,2	15	7,5	4,5
			Development resources		
2,0			0		48



Each Team is driven by Accepted Objectives

Treportal E-Sat Team Objectives for 12 weeks

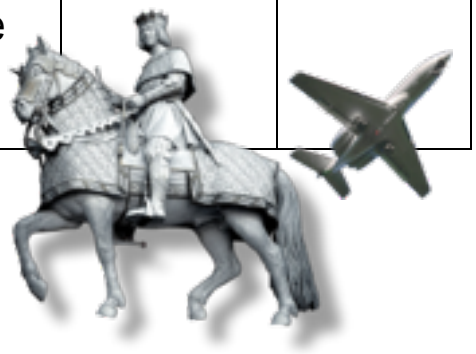
Confirmit Evo Weekly Value Delivery Cycle

	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

Evo's impact on Confirmit product qualities 1st Qtr

- Only 5 highlights of the 25 impacts are listed here

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
Performance.Runtime.Concurrency: Maximum number of simultaneous respondents executing a survey with a click rate of 20 sec and an response time<500 ms, given a defined [Survey-Complexity] and a defined [Server Configuration, Typical]	250 users	6000



confirmit

Release 8.5

Initial Experiences and conclusions

- **EVO has resulted in**
  - **increased motivation and**
  - **enthusiasm amongst developers,**
  - **it opens up for empowered creativity**

- **Developers**
  - **embraced the method and**
  - **saw the value of using it,**
  - **even though they found parts of Evo difficult to understand and execute**

confirmit

Trond Johansen

Conclusions -

- **The method's positive impact on Confirmit produc qualities has convinced us that**
  - **Evo is a better suited development process than o 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 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 knowledge gained from previous steps.**

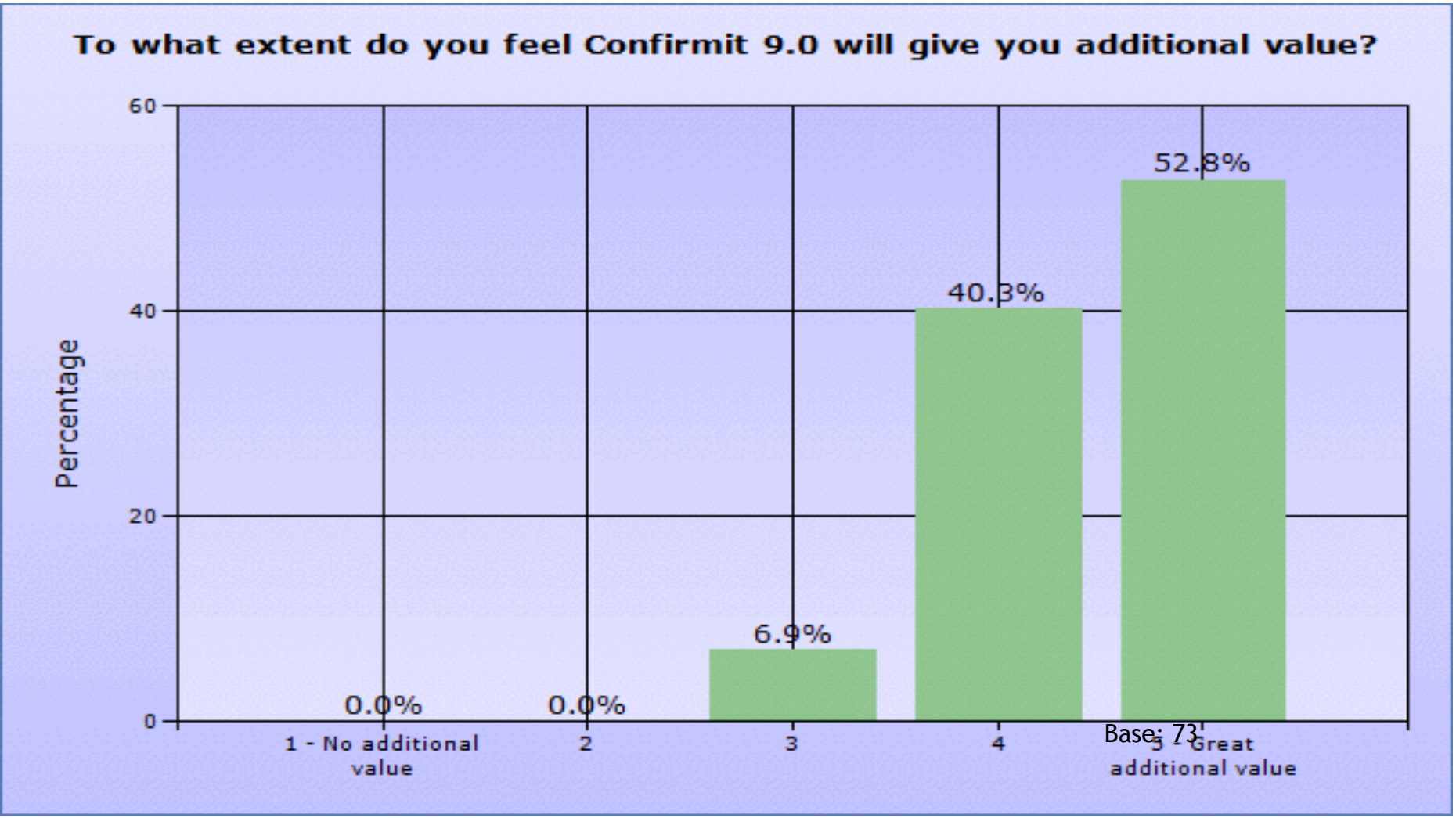
- **The method has**
  - **high focus on measurable product qualities, and**
    - **defining these clearly and testably, requires training and maturity.**
  - **It is important to believe that everything can be measured,**
    - **and to seek guidance if it seems impossible.**

Trond Johansen

Initial Customer Feedback on the new Confirmit 9.0

November 24th, 2004

Initial perceived value of the new release (Base 73 people)



ACTUAL RESULTS IN SECOND 12 WEEKS OF USING THE NEW METHOD

Evo's impact on Confirmit 9.0 product qualities

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%**

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%

MORE ACTUAL RESULTS IN SECOND 12 WEEKS OF USING THE NEW METHOD

Evo's impact on Confirmit 9.0 product qualities

Product quality	Description	Customer value
Performance	Max 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 acceptable.	Number of panelists increased by **1500%**
Scalability	Ability to accomplish a bulk-update of X panelists within a timeframe of Z sec.	Number of panelists increased by **700%**
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%**

The GREEN WEEK: Agile *Technical Debt* **Engineering** beats 'Refactoring'

Tom Gilb
Tom @ Gilb . Com
www.Gilb.com

10 Minute Lightning Talk, 5 Nov 2013

Smidig 2013

Technical debt

consequences of poor software architecture and software development within a codebase.

Causes of technical debt include

① **Business pressures**
② **Lack of process or understanding**
③ **Lack of building loosely coupled components**,
④ **Lack of test suite**,
⑤ **Lack of documentation**,
⑥ **Lack of collaboration**
⑦ **Parallel**
⑧ **Delayed Refactoring**

Conventional Refactoring

	Technique	Description
1	Code Refactoring (clean-up)	It is intended to remove the unused code, methods, variables etc. which are misleading.
2	Code Standard Refactoring	It is done to achieve quality code.
3	Database Refactoring (clean-up)	Just like code refactoring, it is intended to clean or remove the unnecessary and redundant data without changing the architecture.
4	Database schema and design Refactoring	This includes enhancing the database schema by leaving the actual fields required by the application.
5	User-Interface Refactoring	It is intended to change the UI without affecting the underlying functionality.
6	Architecture Refactoring	It is done to achieve modularization at the application level.

Refactoring – to Sustain Application Development Success in Agile Environme
by Narayana Maruvada
In agilerecord.com Nov 1 2013

Impact Software Qualities

- "Importantly, the underlying objective behind refactoring is to give thoughtful consideration and **improve** some of the **essential <*Quality*> *attributes*** of the software."

efactoring – to Sustain Application Development Success in Agile Environments
by Narayana Maruvada
In AGILERECORD.COM NOVEMBER 1 2013

Impact Software Qualities

"Key Benefits of Refactoring

From a system/application standpoint, listed below are summaries of the key benefits that can be achieved seamlessly when implementing the refactoring process in a disciplined fashion:

① Firstly, it improves the overall software **extendability**.

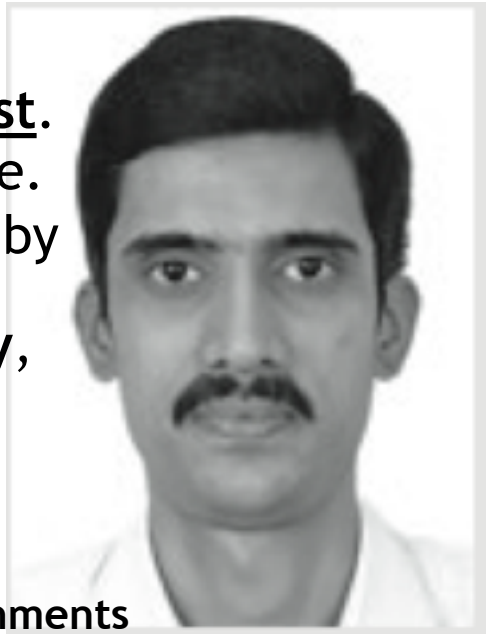
② Reduces and optimizes the code **maintenance cost**.

③ Facilitates highly standardized and organized code.

④ Ensures that the system architecture is improved by retaining the behavior.

⑤ Guarantees three essential attributes: **readability**, **understandability**, and **modularity** of the code.

⑥ Ensures constant improvement in the **overall quality** of the system. "



efactoring – to Sustain Application Development Success in Agile Environments

by Narayana Maruvada

In agilerecord.com Nov 1 2013

Impact Software Qualities

"Key Benefits of Refactoring

From a sys[tem/application standpoint, listed below] are summaries seamlessly [integrated] ss in a disciplin[e]

① First [improves] **exten[sibility]**

② Redu[ces] **cost**.

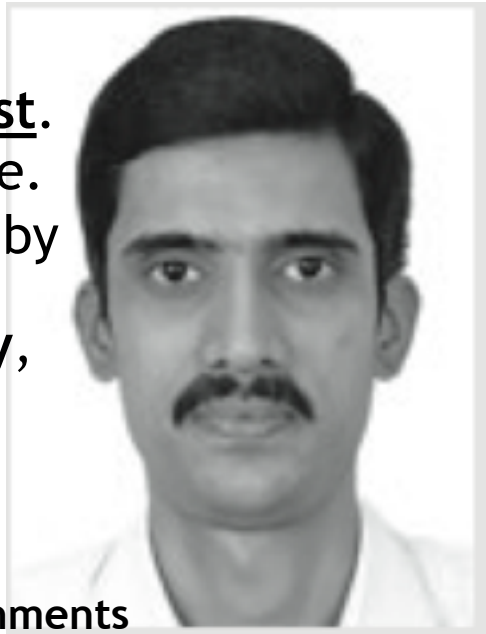
③ Facil[itates] ode.

④ Ensu[res] ed by retain[ing]

⑤ Guarantees three essential attributes: **readability**, **understandability**, and **modularity** of the code.

⑥ Ensures constant improvement in the **overall quality** of the system. "

No numbers given to support this



efactoring - to Sustain Application Development Success in Agile Environments

by Narayana Maruvada

In agilerecord.com Nov 1 2013

There is a smarter way

- But it means we have to become **real** software *engineers*,

- Not just- - - *softcrafters**

- * coders, devs, programmers.
  - Term coined in
  - "Principles of Software Engineering Management", 1988, Gilb

A bright idea: based on experience

- So, Confirmit was getting amazing results for the user, customer, and system level attributes they targeted
- And someone on the team realized…
  - What about us devs and testers
  - We are stakeholders too!
  - Refactoring (1 day a week) was NOT working well.
- Let us try to engineer the qualities that we need into the system
- The same way we engineer the user qualities into the system

Code quality – ”green” week, 2005
“Refactoring by Proactive Design Engineering!”

- **In these ”green” weeks, some of the deliverables will be less visible for the end users, but more visible for our QA department.**
- **We manage code quality through an Impact Estimation table. T.I.**

Current Status		Improvement	Goals			Step 6 (week 14)		Step 7 (wee
	Units		Past	Tolerable	Goal	Estimated Impact	Actual Impact	Estimated Impact
	100,0	100,0	0	80	100			100
		Speed						
	100,0	100,0	0	80	100	100	100	
		Maintainability.Doc.Code						
	100,0	100,0	0	80	100	100	100	
		InterviewerConsole						
		NUnitTests						
	0,0	0,0	0	90	100			
		PeerTests						
	100,0	100,0	0	90	100			100
		FxCop						
	0,0	10,0	10	0	0			
		TestDirectorTests						
	100,0	100,0	0	90	100			100
		Robustness.Correctness						
	2,0	2,0	0	1	2	2	2	
		Robustness.BoundaryConditions						
	0,0	0,0	0					
		Speed						
	0,0	0,0	0					
		ResourceUsage.CPU						
	100,0	0,0	100					
		Maintainability.Doc.Code						
	100,0	100,0	0					
		SynchronizationStatus						
		NUnitTests						

- Speed
- Maintainability
- Nunit Tests
- PeerTests
- TestDirectorTests
- Robustness.Correctness
- Robustness.Boundary Conditions
- ResourceUsage.CPU
- Maintainability.DocCode
- SynchronizationStatus

The Monthly ‘Green Week’

User Week 1
- Select a Goal
- Brainstorm Designs
- Estimate Design Impact

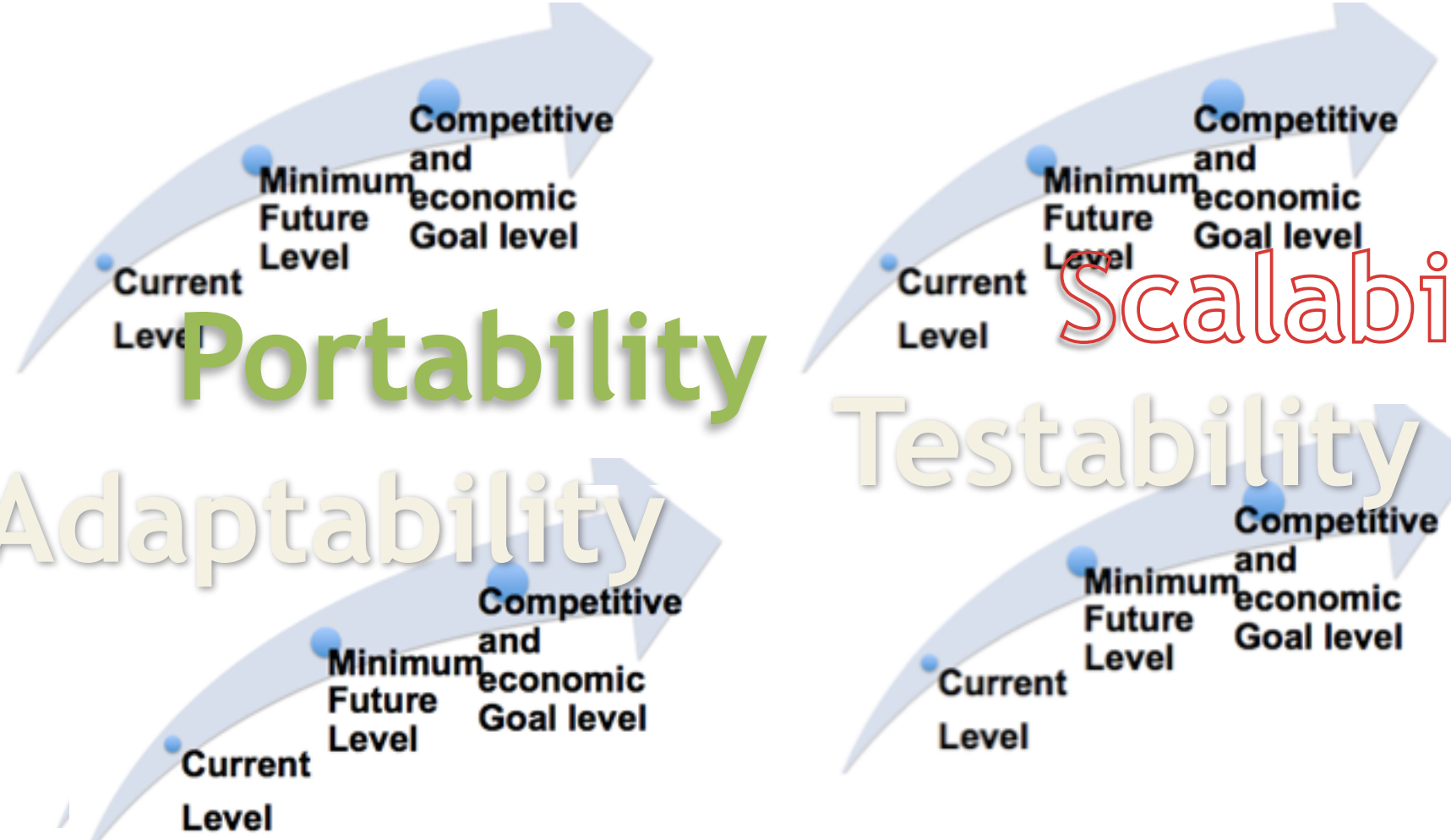
User Week 2
- Select a Goal
- Brainstorm Designs
- Estimate Design Impact

User Week 3
- Select a Goal
- Brainstorm Designs
- Estimate Design Impact

Developer Week 4
- **Select a Goal**
- **Brainstorm Designs**
- **Estimate Design Impact/**

Raising the Levels of Maintainability like ‘Mean Time To Fix a Bug’

Raising the Levels of Maintainability Multiple Attributes of Technical Debt



Broader 'Maintainability' Concepts
ALL *quantified*, with a defined Scale of measure in CE-5

- Performance
  - Quality
    - Availability
      - Reliability
      - Maintainability
      - Integrity
        - Threat
        - Security
    - Adaptability
      - Flexibility
        - Connectability
        - Tailorability
          - Extendibility
          - Interchangeability
      - Upgradeability
        - Installability
        - Portability
        - Improveability

1. The Conscious Design Principle:

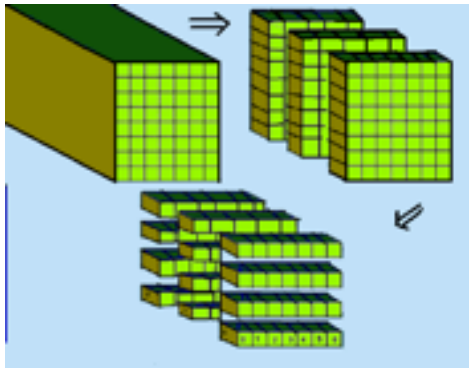
- "Maintainability must be *consciously* designed into a system:
  - failure to **design** to a set of levels of maintainability
  - means the **resulting maintainability** is both *bad* and *random*. "
- © Tom Gilb (2008, INCOSE Paper)
  - http://www.gilb.com/tiki-download_file.php?fileId=138

The 'Maintainability' Generic Breakdown into Sub-problems

1. Problem Recognition Time.
   How can we reduce the time from bug actually occurs until it is recognized and reported?
2. Administrative Delay Time:
   How can we reduce the time from bug reported, until someone begins action on it?
3. Tool Collection Time.

How can we reduce the time delay to collect correct, complete and updated information to analyze the bug: source code, changes, database access, reports, similar reports, test cases, test outputs.

4. Problem Analysis Time.
   Etc. for all the following phases defined, and implied, in the Scale scope above.
5. Correction Hypothesis Time
6. Quality Control Time
7. Change Time
8. Local Test Time
9. Field Pilot Test Time



10. Change Distribution Time
11. Customer Installation Time
12. Customer Damage Analysis Time
13. Customer Level Recovery Time
14. Customer QC of Recovery Time

Source: Competitive Engineering Ch 5
& Ireson (ed.) Reliability Handbook, 1

An Example of Specifying 1 Attribute in 'Planguage'

***Restore Speed**:*

***Type**: Software Quality Requirement.* ***Version**: 25 October 2007.*

***Part of**: Rock Solid Robustness*

***Ambition**: Should an error occur (or the user otherwise desire to do so), the system shall be able to restore the system to a previously saved state in less than 10 minutes. <-6.1.2 HFA.*

Scale: Duration from Initiation of Restore to Complete and verified state of a defined [Previous: Default = Immediately Previous]] saved state.

Initiation: defined as {Operator Initiation, System Initiation, ?}. Default = Any.

Goal [Initial and all subsequent released and Evo steps] 1 minute?

Fail [Initial and all subsequent released and Evo steps] 10 minutes. <- 6.1.2 HFA

Catastrophe: 100 minutes.

Further Reading: Green Week

http://www.gilb.com/dl575, http://www.gilb.com/dl660

Gilb's Mythodology Column

The Green Week: Reducing Technical Debt by Engineering

by Tom & Kai Gilb

Case: Multinational Bank 2011

Critical Project Objectives ‘not clear’

What about You ?

20 Sept, 2011 Report on Gilb Evo method (Richard Smith, Citigroup)

- http://rsbatechnology.co.uk/blog:8
- Back in 2004, I was employed by a large investment bank in their FX e-commerce IT department as a business analyst.
- The wider IT organisation used a complex waterfall-based project methodology that required use of an intranet application to manage and report progress.
- However, it's main failings were that it almost **totally missed the ability to track delivery of actual value improvements to a project's stakeholders**, and **the ability to react to changes in requirements and priority for the project's duration.**
- The toolset generated lots of charts and stats that provided **the illusion of risk control**. but actually provided very little help to the analysts, developers and testers actually doing the work at the coal face.
- The proof is in the pudding;
  - I have **used Evo** (albeit in disguise sometimes) on two large, high-risk projects in front-office investment banking businesses, and several smaller tasks.
  - On the largest critical project, the original business functions & performance objective **requirements document, which included no design, essentially remained unchanged** over the 14 months the project took to deliver,
  - but the detailed **designs** (of the GUI, business logic, performance characteristics) **changed many many times**, guided by lessons learnt and **feedback** gained by delivering a succession of early deliveries to real users.
  - In the end, the new system responsible for 10s of USD billions of notional risk, **successfully went live over one weekend for 800 users worldwide**, and **was seen as a big success by the sponsoring stakeholders.**

" I attended a 3-day course with you and Kai whilst at Citigroup in 2006"

Richard Smith

“ I attended a 3-day course with you and Kai whilst at Citigroup in 2006”

Previous PM Methods:
No ‘Value delivery tracking’.
No change reaction ability

Richard Smith

- “However, (**our old** project management methodology) main failings were that
- it almost **totally missed the ability to track delivery of actual *value* improvements to a project's stakeholders**,
- and **the ability to react to changes**
  - **in requirements and**
  - **priority**
  - **for the project's duration”**

We only had the illusion of control. But little help to testers and analysts

Richard Smith

- "The (old) toolset generated lots of charts and stats
- that provided **the illusion of risk control**.
- But actually provided very little help to the analysts, developers and testers actually doing the work at the coal face."

The proof is in the pudding;

Richard Smith

- "The proof is in the pudding;
- I have **used Evo**
  - *(albeit in disguise sometimes)*
  - on two large, high-risk projects in front-office investment banking businesses,
  - and several smaller tasks. "

Experience: if top level requirements are *separated* from design, the 'requirements' are **stable**!

Richard Smith

- "On the largest critical project,
- the original ***business functions & performance objective*** **requirements document,**
- ***which included*** **no design,**
- essentially remained ***unchanged***
- over the **14 months** the project took to deliver,…."

" I attended a 3-day course with you and Kai whilst at Citigroup in 2006", Richard Smith

Dynamic (Agile, Evo) design testing: not unlike 'Lean Startup'

Richard Smith

- "… but the detailed **designs**
  - **(of the GUI, business logic, performance characteristics)**
- **changed** many many times,
- guided by lessons learnt
- and **feedback** gained by
- delivering a succession of early deliveries
- to real users"

" I attended a 3-day course with you and Kai whilst at Citigroup in 2006", Richard Smith

It looks like the stakeholders liked the top level system qualities, on first try

Richard Smith

- " In the end, the new system responsible for 10s of USD billions of notional risk,
- **successfully went live**
- **over one weekend**
- **for 800 users worldwide**,
- and **was seen as a big success**
- **by the sponsoring stakeholders."**

" I attended a 3-day course with you and Kai whilst at Citigroup in 2006" , Richard Smith

Bank Training like Richard Used

THE LEARNING PROCESS

THEORY, PRACTICE, DISCUSS, DOCUMENTATIONS

1. Lectures (50%)

Basic Theory (Principles, Standards, Rules, Templates)
Case studies (as far as possible from DB and banking)
Examples of practice (as far as possible from DB and banking)

2. Questions and discussion

3. Participant exerci
(small groups 2 to 4), followed up by Instructors, and experienced DB assistants (if available)

4. Substantial digital documentation, a library of books, papers, cases

Requirements Course Outline http://www.gilb.com/dl522

Day 1
Quantify Requirements

1. **Overview**: Evo & Planguage in relation to Agile

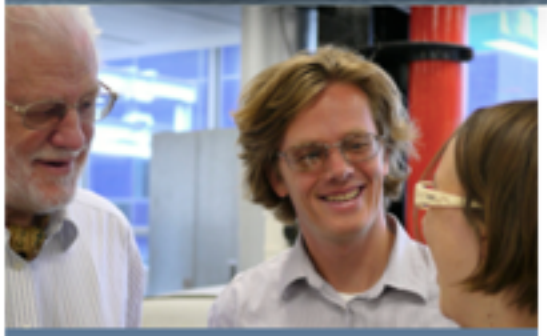
Day 2
Standards, Principles, Risks

1. Tips for **analyzing** project plans to find the 'real' value

Day 3
Design, Delivery, Culture Change

1. **estimating** the quantified impact of a **design** on

Requirements Workshop



WORKSHOP ADVANTAGES

a complete method for tackling all the critical and real stakeholder requirements for a project, at all levels of consideration for IT Projects.

BAR NONE
the most advanced and comprehensive workshop on requirements specification in the world.

Master how you communicate your organisation's 'real' requirements, and your stakeholders' most critical improvement requirements, in an unambiguous, clear, measurable, and testable way.

Project and System Level Requirements Specifications

Workshop Objectives:

This workshop will allow you to walk away with practical ability to improve your projects most critical requirements.

You will be able to identify, classify and specify critical project and stakeholder

Workshop Intended for:

People who write requirements (BAs), and their managers. Product owners, project managers and their managers Consultants, engineering/IT methods owners and teachers.

Workshop

Requirements Course Outline http://www.gilb.com/dl522

Detailed Syllabus: Metrics for a bank

Day 1
Quantify Requirements

1. **Overview**: Evo & Planguage in relation to Agile Methods

2. practical **examples** of Planguage for requirements (case studies)

3. the various requirements **concepts** defined deeply and exemplified

4. requirements **templates** (to make standards practical) design constraint templates (a type of required design or architecture)

5. how to **quantify** any qualitative requirement (like intuitiveness or adaptability or security) – this is the key ability that most all other 'requirements' workshops do not teach!

6. **advanced** scale of measure specification methods (a 'scale' is more than units)

Day 2
Standards, Principles, Risks

1. Tips for **analyzing** project plans to find the 'real' value requirements.

2. **standards** for requirements (rules, processes, templates, glossary)

3. **principles** for requirements (help you to tackle new problems better)

4. **quality control** of requirements: measuring requirement conformance to standards (reviews, inspections, agile reviews)

5. how to give information that determines **priorities** of requirements (example Wish/ Goal/Fail and Qualifiers)

6. how to include requirement information about **risks and uncertainties**

Day 3
Design, Delivery, Culture Change

1. **estimating** the quantified impact of a **design** on requirements

2. evolutionary project management and how it integrates with requirements. The **Evo cycle** and how it relates to Agile iteration.

3. **training** requirements writers: how to train colleagues and yourself

4. changing requirements **culture**: how to change your culture of requirements

5. expected **results** from requirements culture improvement: how to measure or know that things are working well

6. a **policy** for improved requirements: summary of main guidelines for value driven projects, and value requirements.

Requirements Course Outline http://www.gilb.com/dl522

End

- Free CE and Evo book
  - Send email to tom@gilb.com
  - Subject ‘Book’