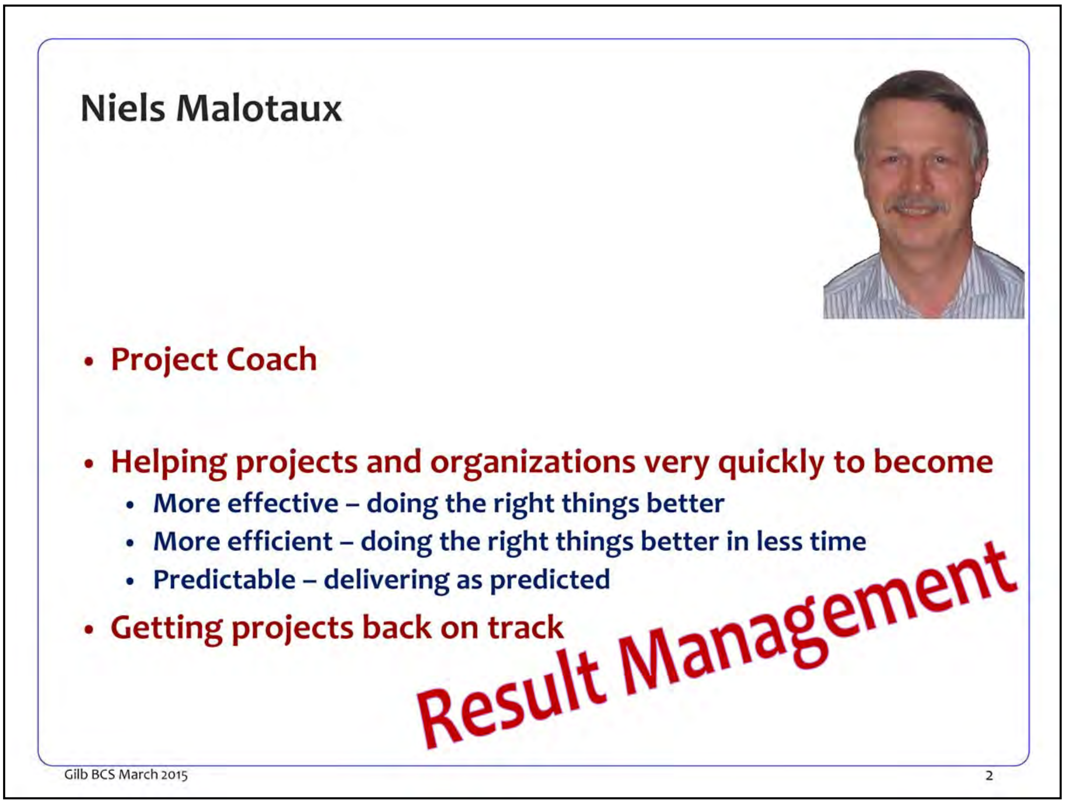
How to get the message across

www.malotaux.nl/?id=conferences

Niels Malotaux

N R Malotaux
Consultancy

+31-655 753 604 niels@malotaux.nl www.malotaux.nl



Niels Malotaux

- Project Coach
- Helping projects and organizations very quickly to become
  - More effective – doing the right things better
  - More efficient – doing the right things better in less time
  - Predictable – delivering as predicted
- Getting projects back on track

Result Management

Niels Malotaux

Graduated **Electronics** at Delft University of Technology in **1974**

Army service at the Dutch Laboratory for Electronic Developments for the Armed Forces, designing computer systems

Philips Electronics – Application support for microcomputer systems design

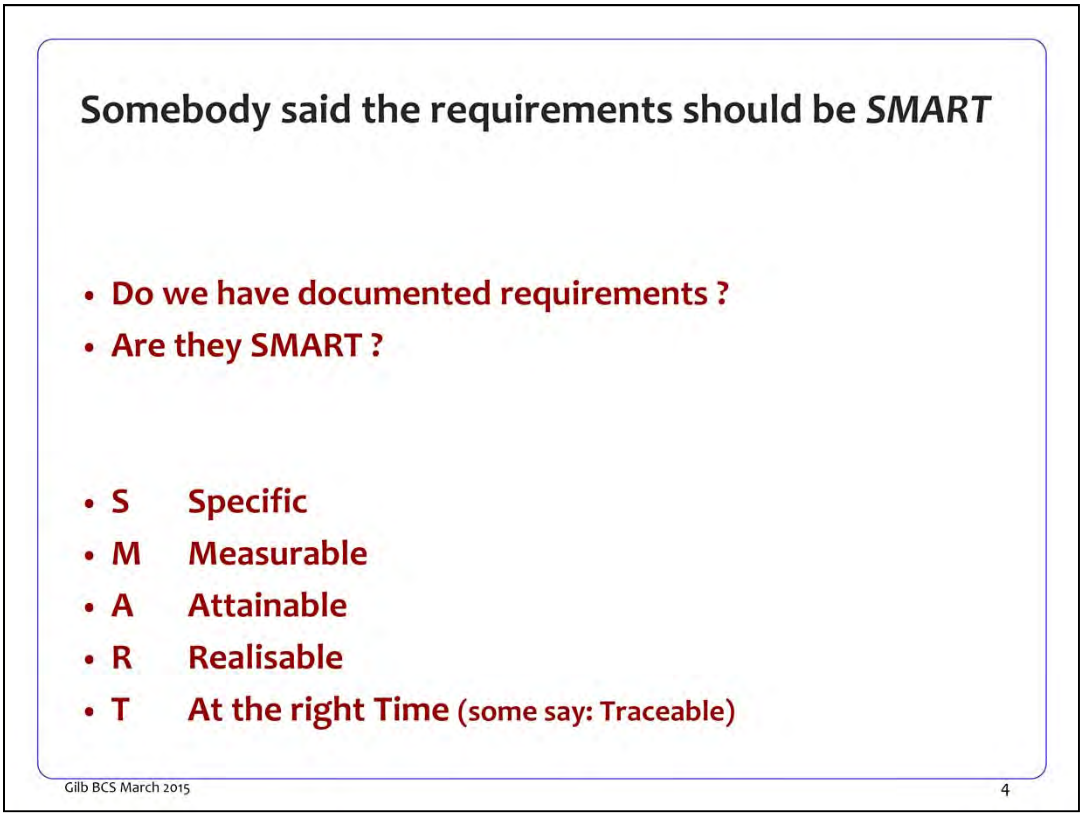
Malotaux - Electronic Systems Design - : developing electronic systems for clients products

Now: **N R Malotaux - Consultancy**

Do engineers really know ?

- **Heathrow Terminal 5: Great success !**
  - **Normal people aren't interested in the technical details of a terminal**
  - **They only want to check-in their luggage as *easily* as possible**

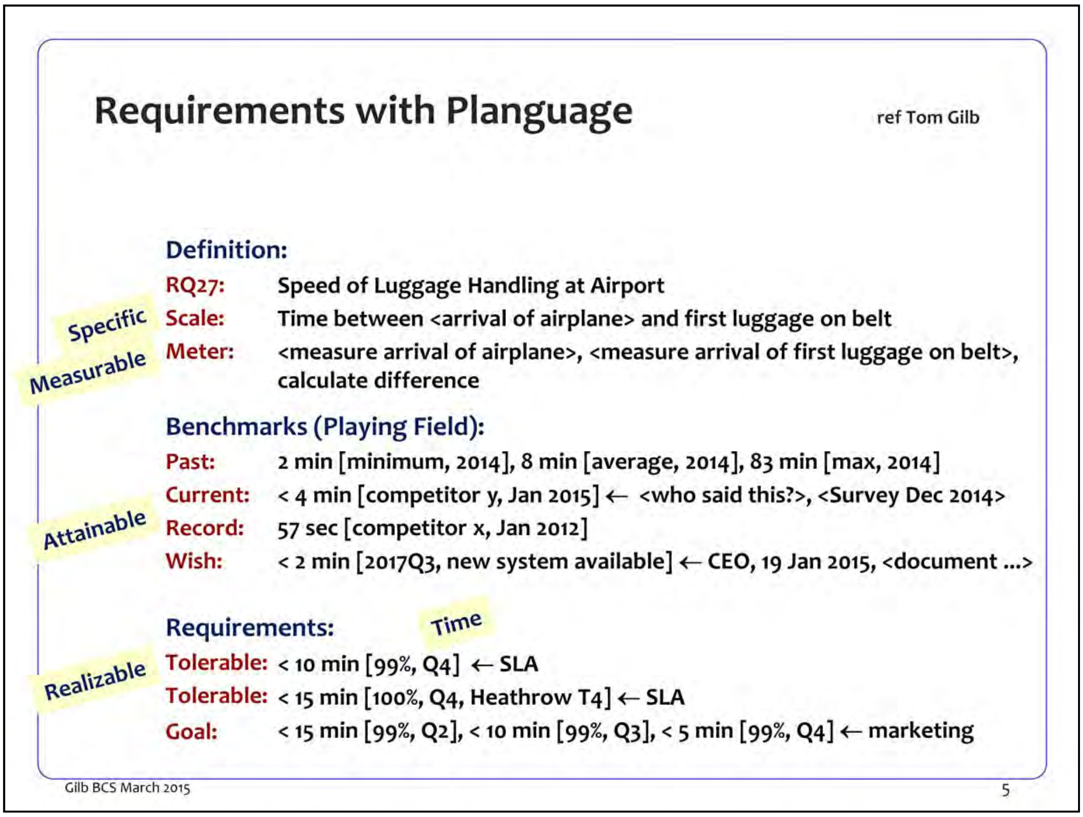
    and
  - **Get their luggage back as *quickly* as possible in *acceptable condition at their destination***
  - **They didn't**
- **One of the problems is to determine what the project** (or your work in general) **really is about**
- **What are the 'real' requirements ?**
- **Clear focus towards the real requirements saves time**



Somebody said the requirements should be *SMART*

- Do we have documented requirements ?
- Are they SMART ?

- S Specific
- M Measurable
- A Attainable
- R Realisable
- T At the right Time (some say: Traceable)



Requirements with Planguage

ref Tom Gilb

Definition:

Specific

Measurable

RQ27: Speed of Luggage Handling at Airport

Scale: Time between <arrival of airplane> and first luggage on belt

Meter: <measure arrival of airplane>, <measure arrival of first luggage on belt>, calculate difference

Benchmarks (Playing Field):

Attainable

Past: 2 min [minimum, 2014], 8 min [average, 2014], 83 min [max, 2014]

Current: < 4 min [competitor y, Jan 2015] ← <who said this?>, <Survey Dec 2014>

Record: 57 sec [competitor x, Jan 2012]

Wish: < 2 min [2017Q3, new system available] ← CEO, 19 Jan 2015, <document ...>

Requirements: Time

Realizable

Tolerable: < 10 min [99%, Q4] ← SLA

Tolerable: < 15 min [100%, Q4, Heathrow T4] ← SLA

Goal: < 15 min [99%, Q2], < 10 min [99%, Q3], < 5 min [99%, Q4] ← marketing

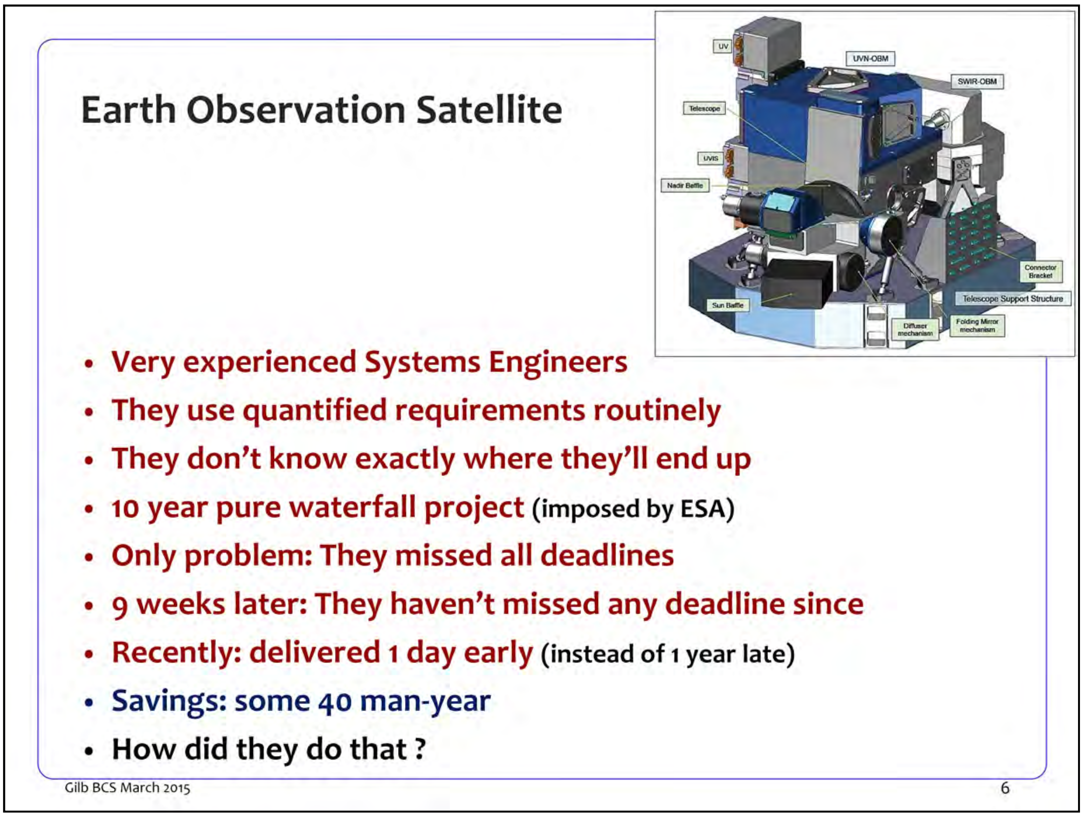
Gilb BCS March 2015

5

This is my standard simple Planguage example. It shows a bit more (Current, Record, Wish) than the bare minimum of a performance requirement.

Someone said that the ‘requirements should be SMART’. Of course Planguage provides SMART requirements.

See www.malotaux.nl/planguage for an explanation of the elements shown.



UV
UVN-OBM
SWIR-OBM
Telescope
UVIS
Nadir Baffle
Connector Bracket
Telescope Support Structure
Sun Baffle
Diffuser mechanism
Folding Mirror mechanism

Earth Observation Satellite

- Very experienced Systems Engineers
- They use quantified requirements routinely
- They don't know exactly where they'll end up
- 10 year pure waterfall project (imposed by ESA)
- Only problem: They missed all deadlines
- 9 weeks later: They haven't missed any deadline since
- Recently: delivered 1 day early (instead of 1 year late)
- Savings: some 40 man-year
- How did they do that ?

I was asked to coach a project where very clever Systems Engineers we developing an earth observation instrument to be launched this year.

These people said to me: "Niels. We are doing this kind of work already for 27 years. We're very good at it. What do you think you can add to that?", which of course was a relevant question.

Well, I didn't have to tell them much about Planguage because seasoned Systems Engineers know how to quantify requirements (example on next slide).

But there was one thing they still hadn't mastered: The missed every deadline and were complaining about impossible deadlines.

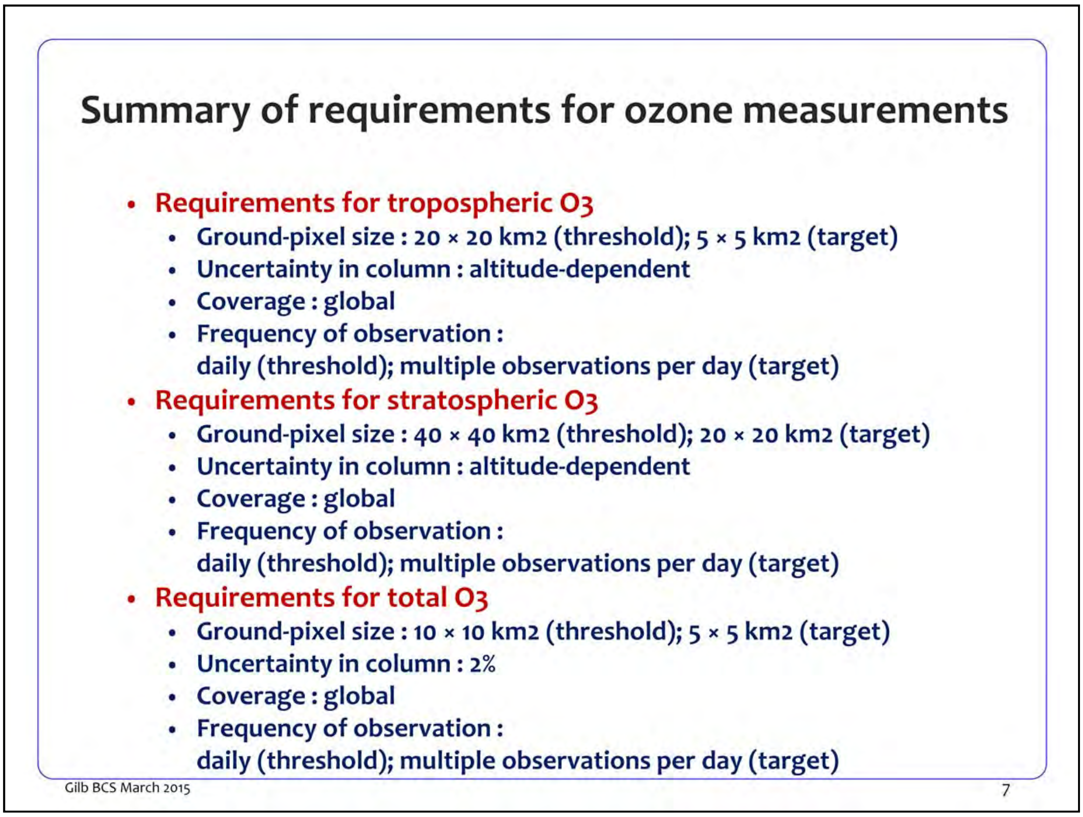
9 weeks later, and ever since they didn't miss any deadline.

I'll explain how they did this.

A few weeks ago I heard that they delivered one day before the expected deadline.

Normally, all their projects take at least one year more than the expected deadline.

Savings: 40 man year.

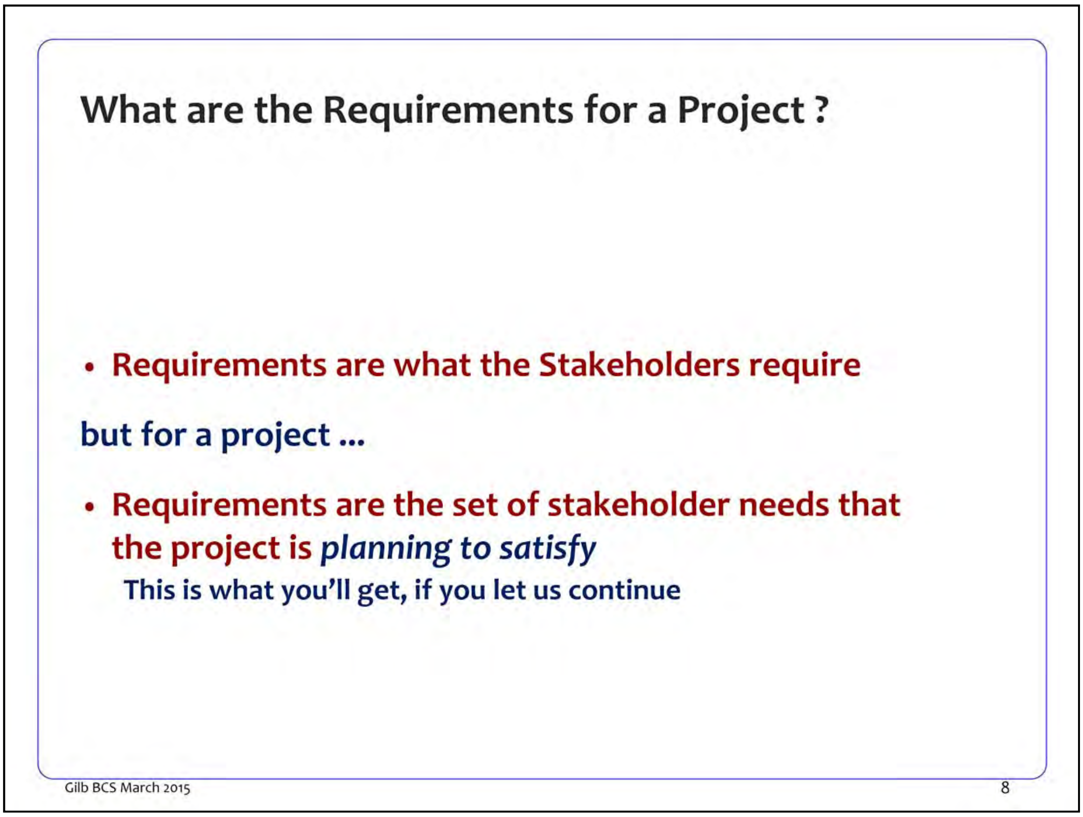


Summary of requirements for ozone measurements

- Requirements for tropospheric O3
  - Ground-pixel size : 20 × 20 km2 (threshold); 5 × 5 km2 (target)
  - Uncertainty in column : altitude-dependent
  - Coverage : global
  - Frequency of observation :
    daily (threshold); multiple observations per day (target)
- Requirements for stratospheric O3
  - Ground-pixel size : 40 × 40 km2 (threshold); 20 × 20 km2 (target)
  - Uncertainty in column : altitude-dependent
  - Coverage : global
  - Frequency of observation :
    daily (threshold); multiple observations per day (target)
- Requirements for total O3
  - Ground-pixel size : 10 × 10 km2 (threshold); 5 × 5 km2 (target)
  - Uncertainty in column : 2%
  - Coverage : global
  - Frequency of observation :
    daily (threshold); multiple observations per day (target)

Gilb BCS March 2015 7

Example of requirements I found on Internet (but cannot find anymore ☹).



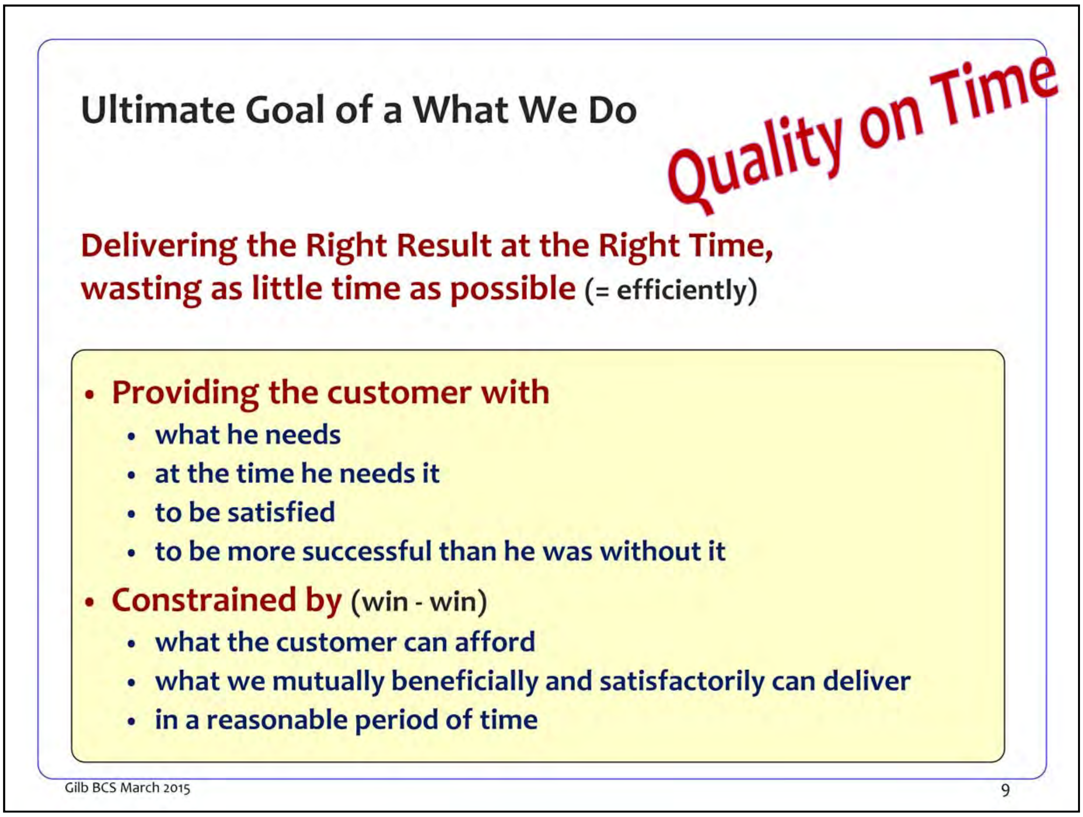
What are the Requirements for a Project ?

- **Requirements are what the Stakeholders require**

but for a project ...

- **Requirements are the set of stakeholder needs that the project is *planning to satisfy***

  This is what you'll get, if you let us continue



Ultimate Goal of a What We Do

Quality on Time

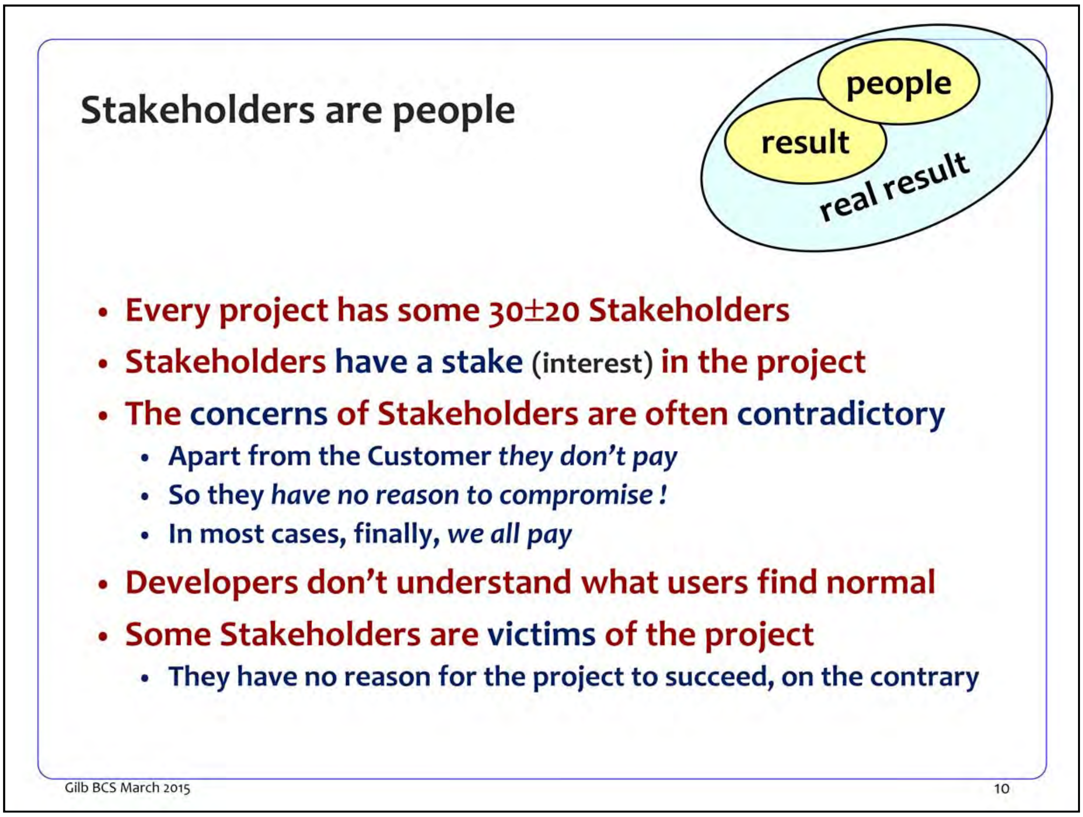
Delivering the Right Result at the Right Time, wasting as little time as possible (= efficiently)

- **Providing the customer with**
  - what he needs
  - at the time he needs it
  - to be satisfied
  - to be more successful than he was without it
- **Constrained by** (win - win)
  - what the customer can afford
  - what we mutually beneficially and satisfactorily can deliver
  - in a reasonable period of time

Gilb BCS March 2015 9

This is to me the top-level requirement for any project or any work we do.

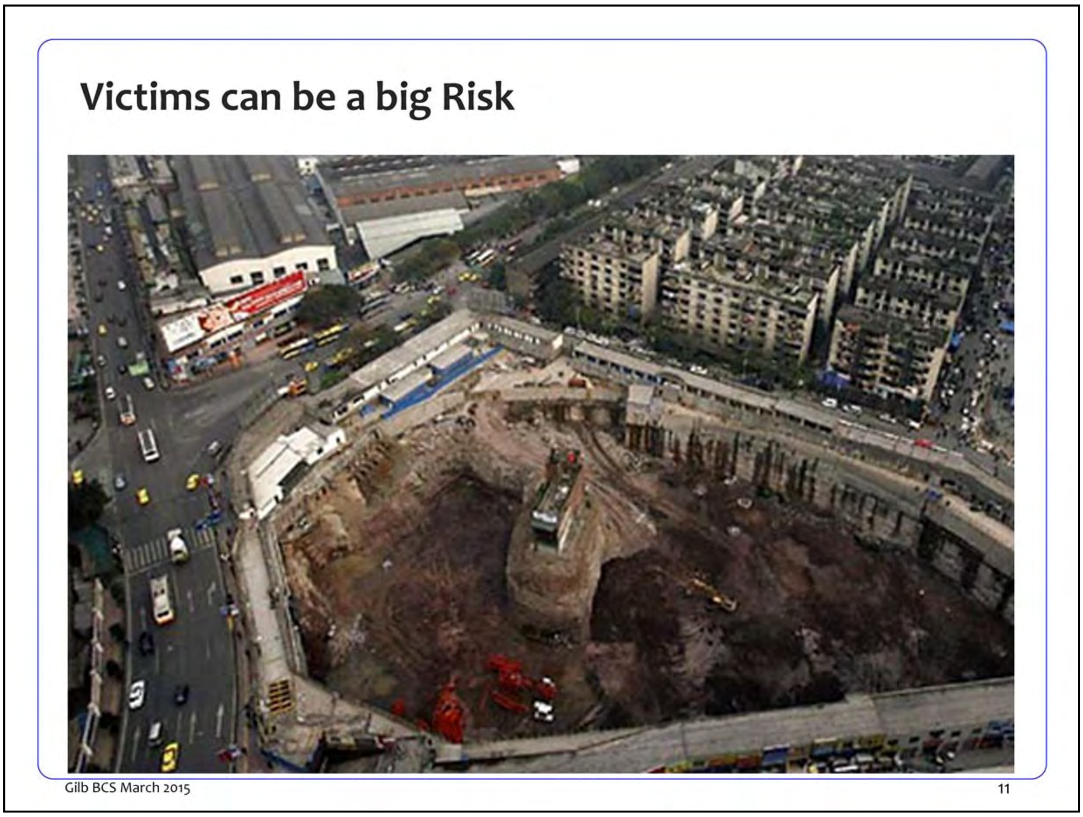
- The customer is the entity that orders and pays. The customer, however, in many cases doesn't use the result of our project himself. He gets the benefit through the users of the result.
- What the customer says he wants is usually not what he really needs
- The time he needs it may be earlier or later than he says
- If the customer isn't satisfied, he doesn't want to pay
- If the customer isn't successful with what we deliver, he cannot pay
- If he's not more successful, why would he pay?
- What the customer wants, he cannot afford. If we try to satisfy all customer's wishes, we'll probably fail from the beginning. We can do great things, given unlimited time and money. But neither the customer nor we have unlimited time and money. Therefore: The requirements are what the Stakeholders require, but for a project: the requirements are what the project is planning to satisfy.
- The customer is king, but we aren't slaves. Both sides should benefit and be happy with the result.
- We will get the best result in the shortest possible time, but not shorter than possible. The impossible takes too much time.



people
result
real result

Stakeholders are people

- Every project has some 30±20 Stakeholders
- Stakeholders have a stake (interest) in the project
- The concerns of Stakeholders are often contradictory
  - Apart from the Customer *they don't pay*
  - So they *have no reason to compromise !*
  - In most cases, finally, *we all pay*
- Developers don't understand what users find normal
- Some Stakeholders are victims of the project
  - They have no reason for the project to succeed, on the contrary



Victims can be a big Risk

Gilb BCS March 2015

11

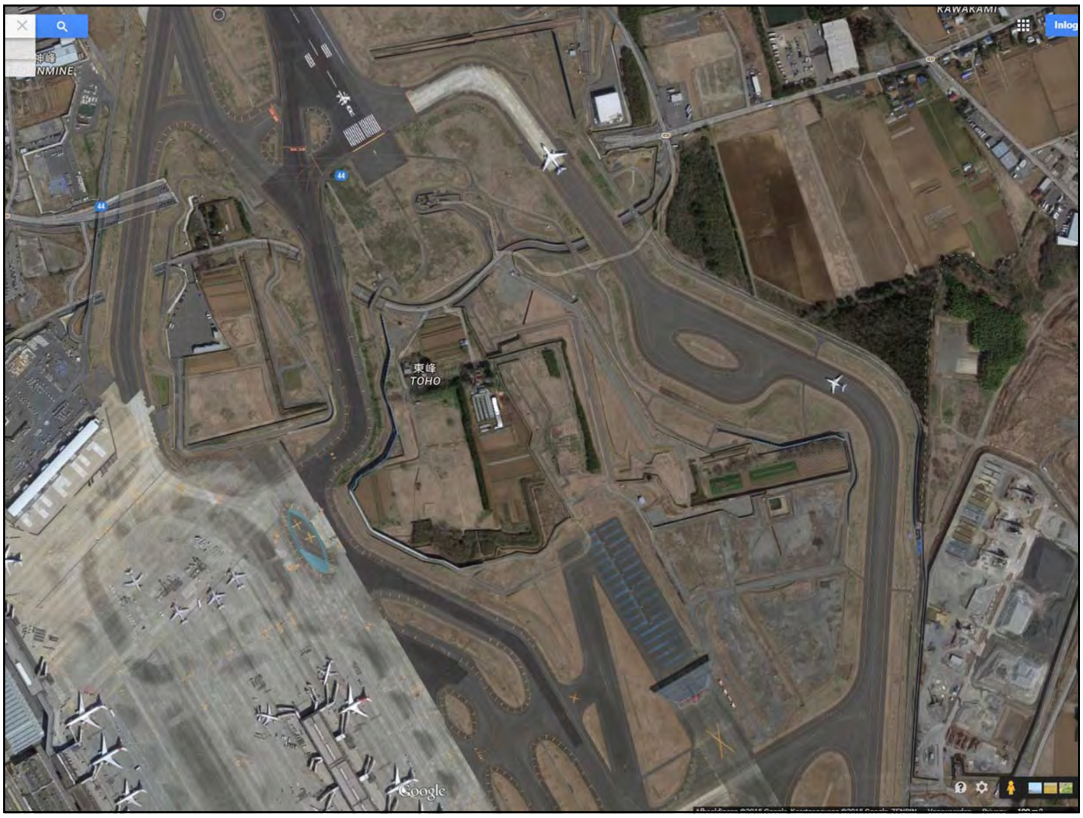
Example of a victim in China

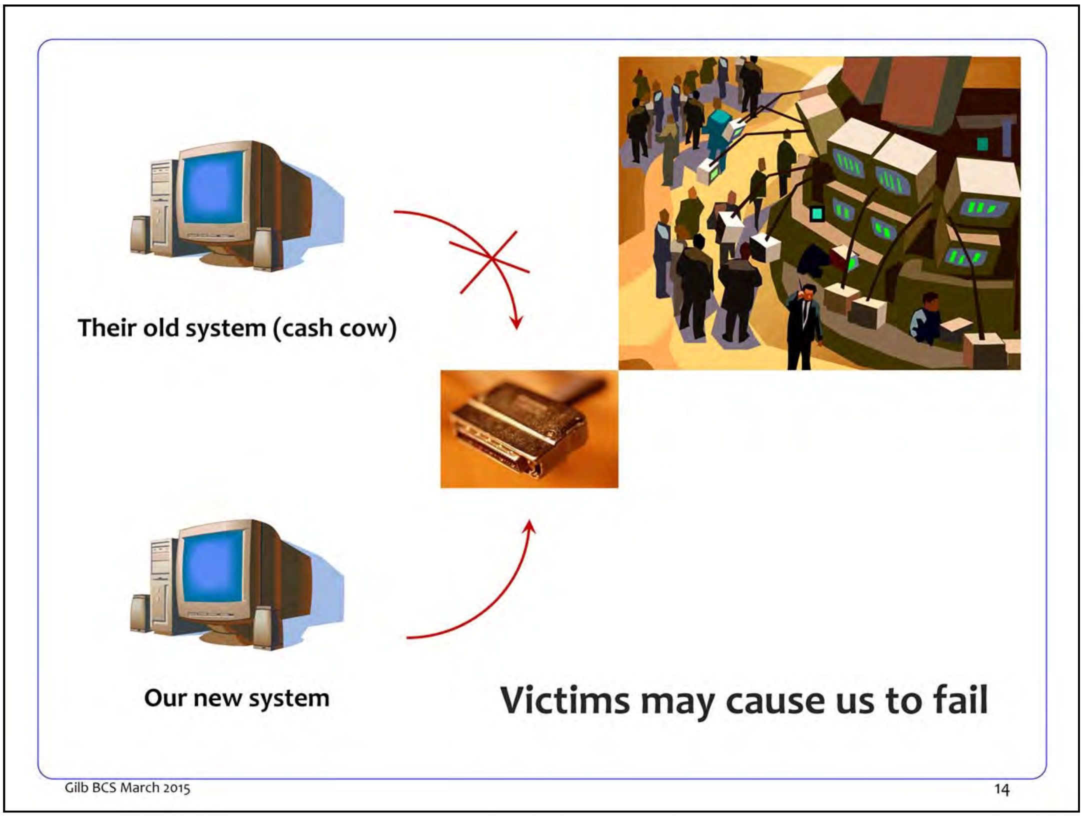
Victims:
Narita Airport

Gilb BCS March 2015

12

Tokyo airport still suffers from victims


峰
NMINE
KAWAKAMI
Inlog
東峰
TOHO
44
Google



Their old system (cash cow)

Our new system

Victims may cause us to fail

Gilb BCS March 2015

14

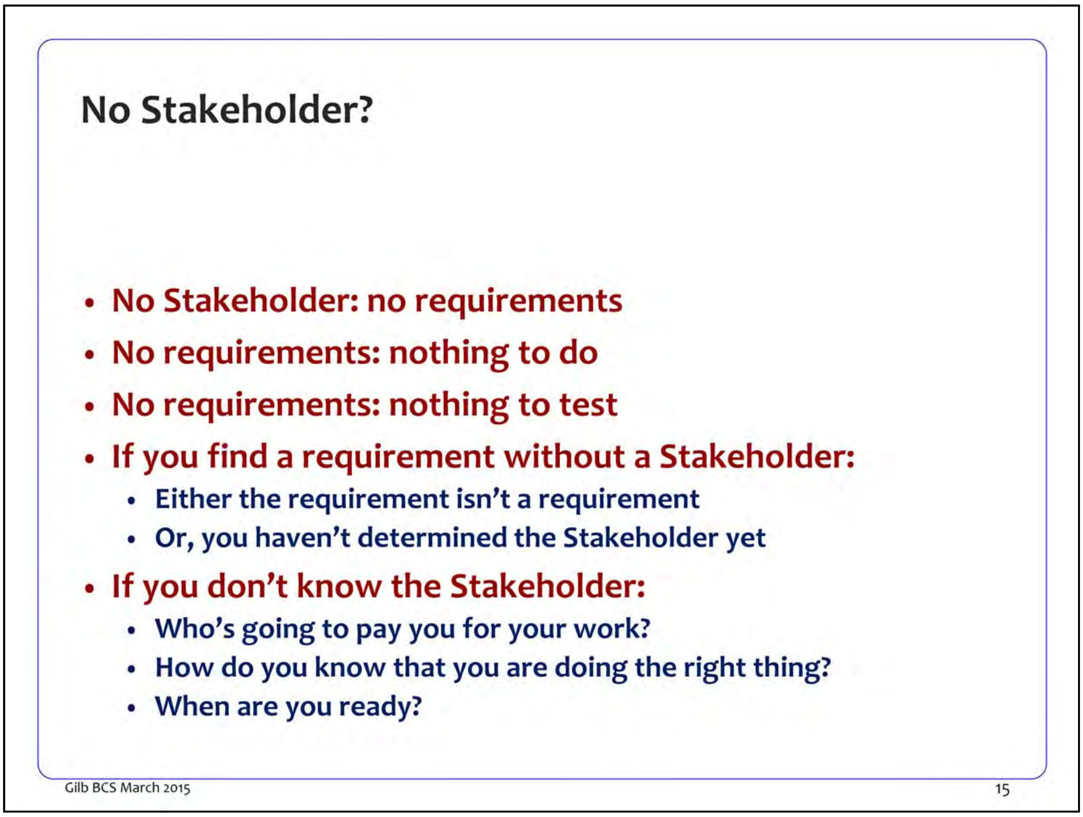
We were developing a system to take over the system of a competitor.

In order to test our system, we needed a test-environment of our competitor.

It was February and we had to deliver in October.

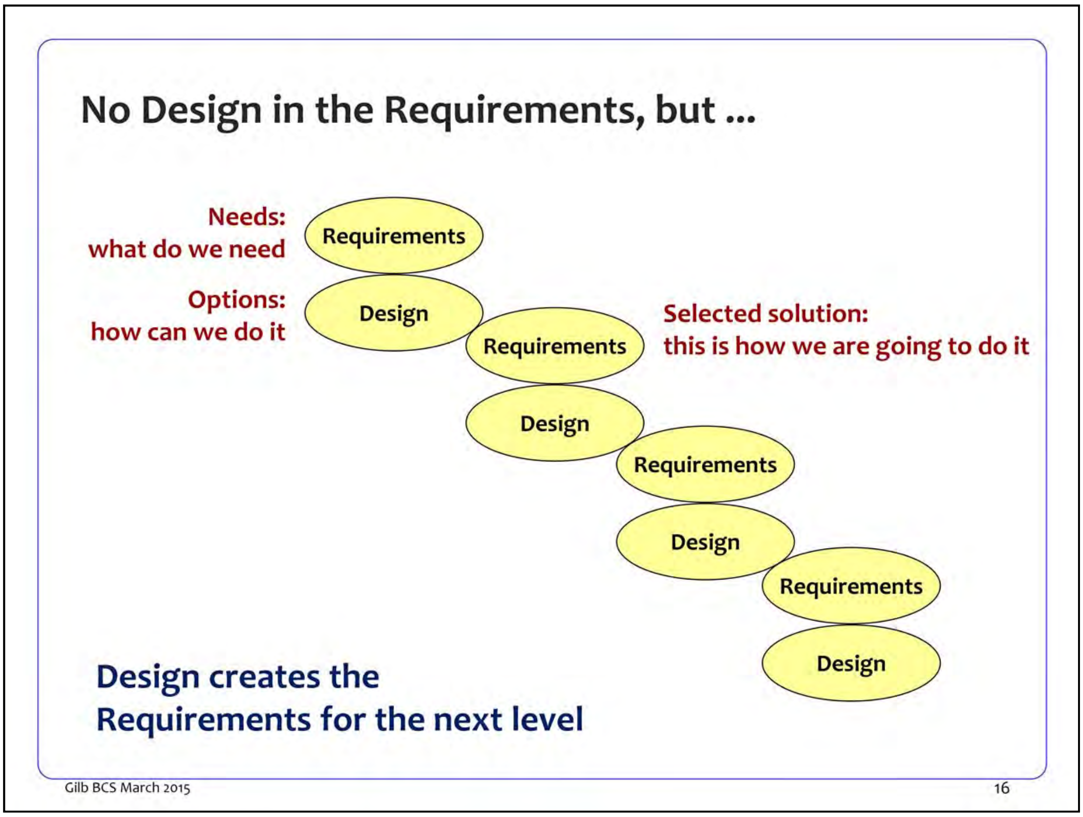
The competitor said: “You can use the test system by August.”

Do you think they really would allow us to test in August?



No Stakeholder?

- No Stakeholder: no requirements
- No requirements: nothing to do
- No requirements: nothing to test
- If you find a requirement without a Stakeholder:
  - Either the requirement isn't a requirement
  - Or, you haven't determined the Stakeholder yet
- If you don't know the Stakeholder:
  - Who's going to pay you for your work?
  - How do you know that you are doing the right thing?
  - When are you ready?



No Design in the Requirements, but ...

Needs:
what do we need

Options:
how can we do it

Selected solution:
this is how we are going to do it

Requirements

Design

Requirements

Design

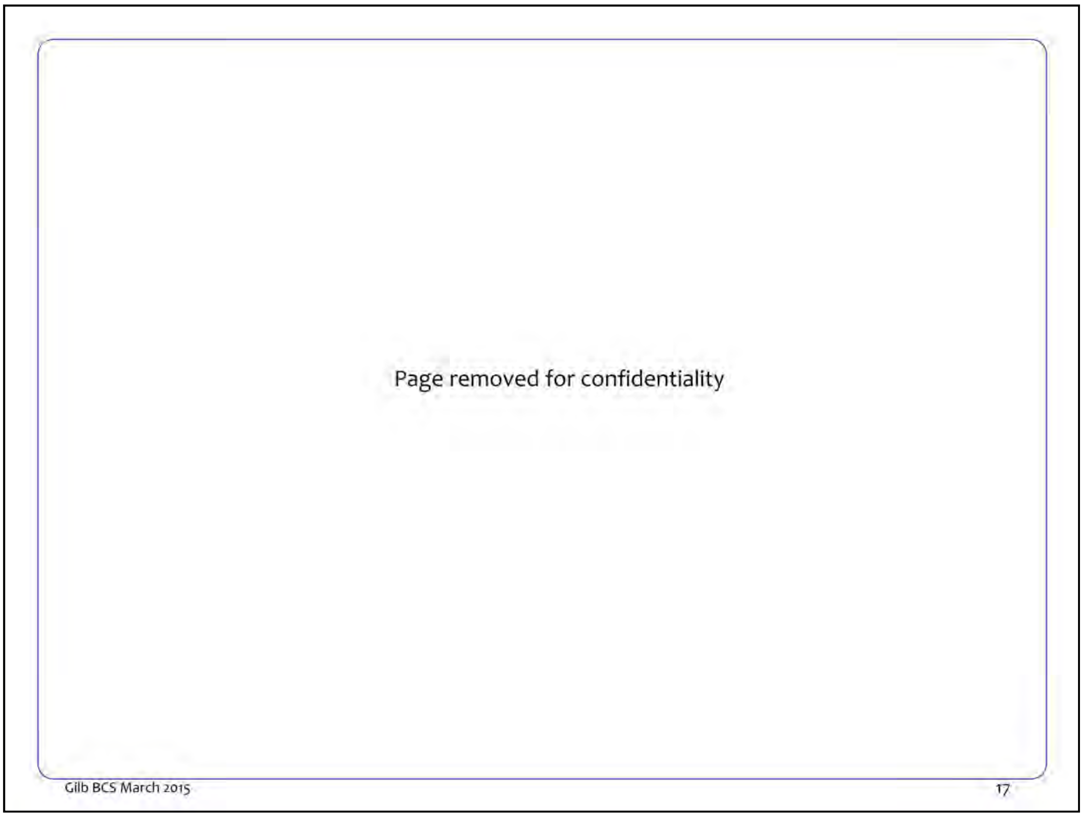
Requirements

Design

Requirements

Design

Design creates the Requirements for the next level



Page removed for confidentiality

Gilb BCS March 2015

17

A case of downloading firmware updates in building automation controllers dispersed in office buildings, hospitals, hotels, supermarket chains etc.

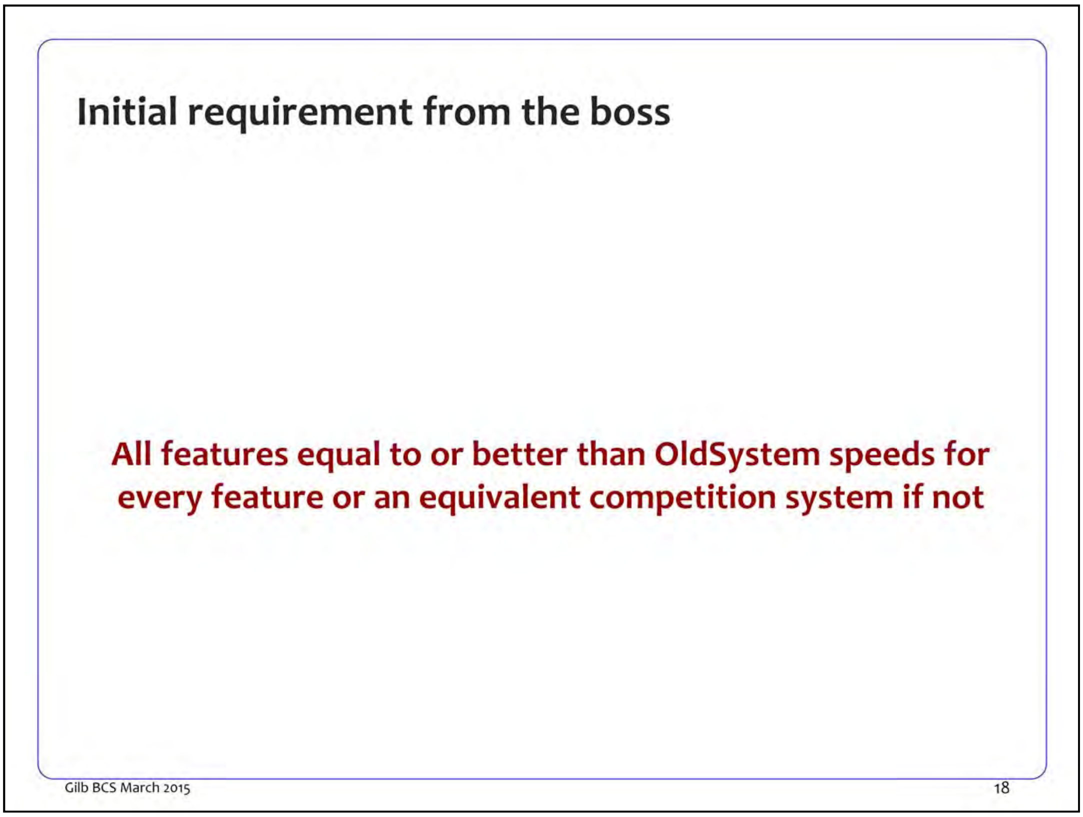
Can be thousands of controllers in one network.

Updating needed the service company to be in the building until all controllers were updated.

The updates were necessary to rectify bugs we caused.

Hence the service companies (our customers) were not amused to waste much time for the update.

Updates did cost too much. How could we improve the download time for many controllers?

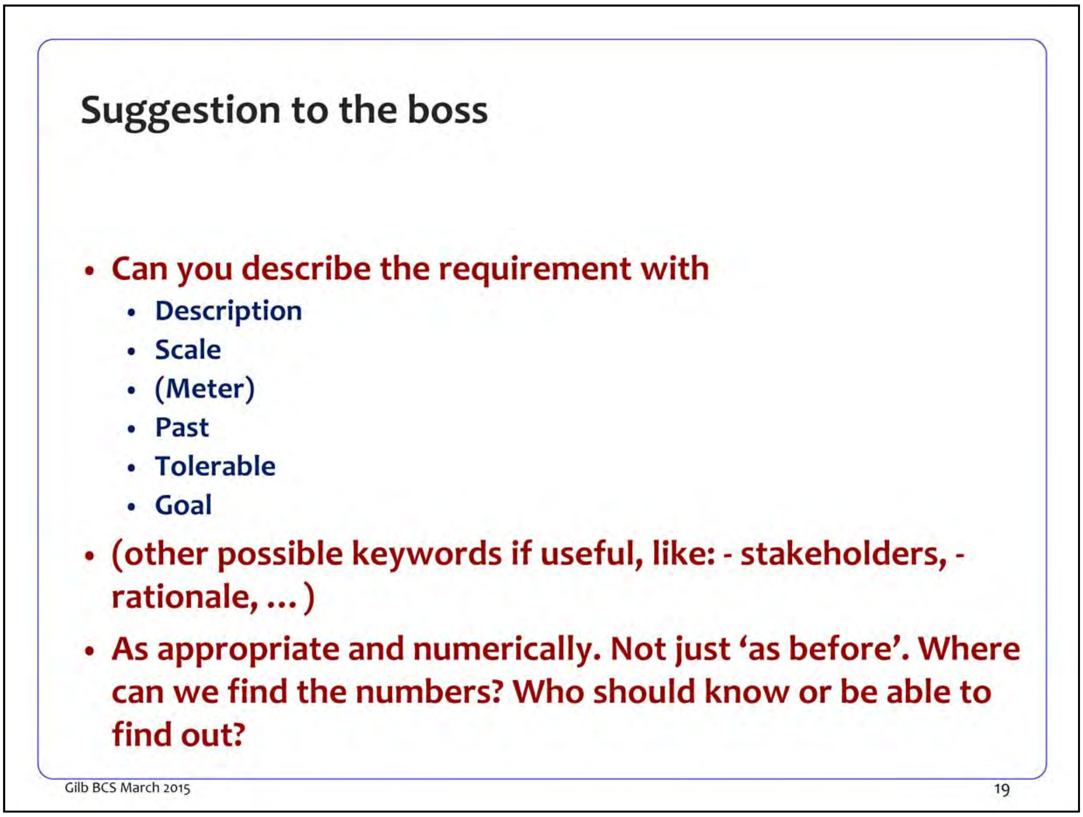


Initial requirement from the boss

All features equal to or better than OldSystem speeds for every feature or an equivalent competition system if not

I asked the boss what his requirements was for the improvement. This was not just about the download speed, but in general a requirement for the next version of the firmware of new types of controllers.

Recognize the management poetry.

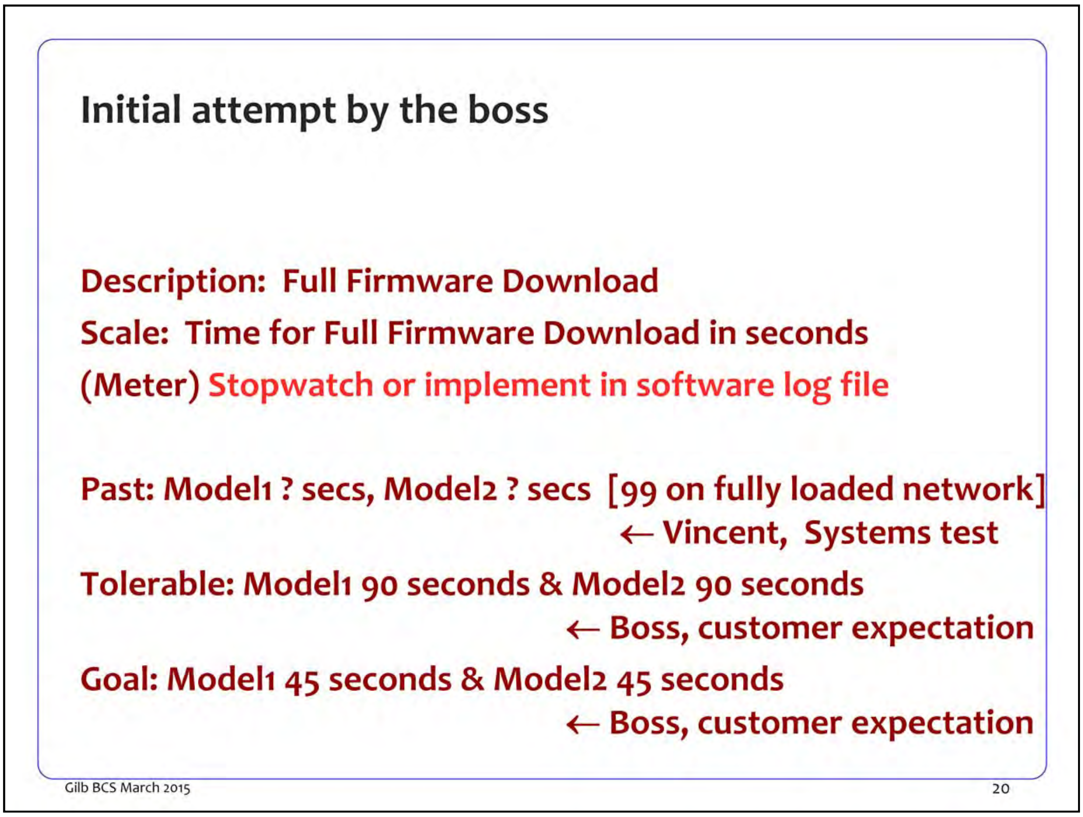


Suggestion to the boss

- Can you describe the requirement with
  - Description
  - Scale
  - (Meter)
  - Past
  - Tolerable
  - Goal
- (other possible keywords if useful, like: - stakeholders, - rationale, …)
- As appropriate and numerically. Not just ‘as before’. Where can we find the numbers? Who should know or be able to find out?

Gilb BCS March 2015 19

I told the boss that there is a way to describe his requirement more concisely, and gave him a hint as shown.



Initial attempt by the boss

Description: Full Firmware Download

Scale: Time for Full Firmware Download in seconds

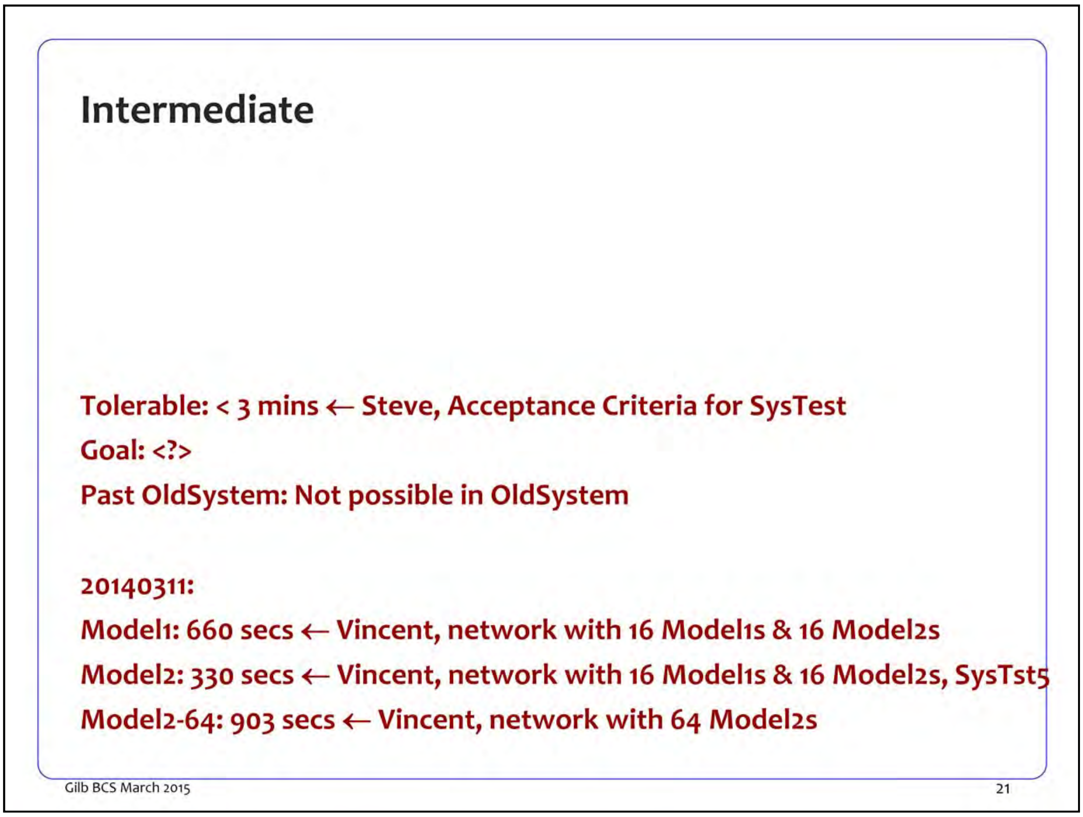
(Meter) Stopwatch or implement in software log file

Past: Model1 ? secs, Model2 ? secs [99 on fully loaded network]
← Vincent, Systems test

Tolerable: Model1 90 seconds & Model2 90 seconds
← Boss, customer expectation

Goal: Model1 45 seconds & Model2 45 seconds
← Boss, customer expectation

A day later he came up with this first go (I added a suggestion for ‘meter’). Vincent was the tester.



Intermediate

Tolerable: < 3 mins ← Steve, Acceptance Criteria for SysTest

Goal: <?>

Past OldSystem: Not possible in OldSystem

20140311:

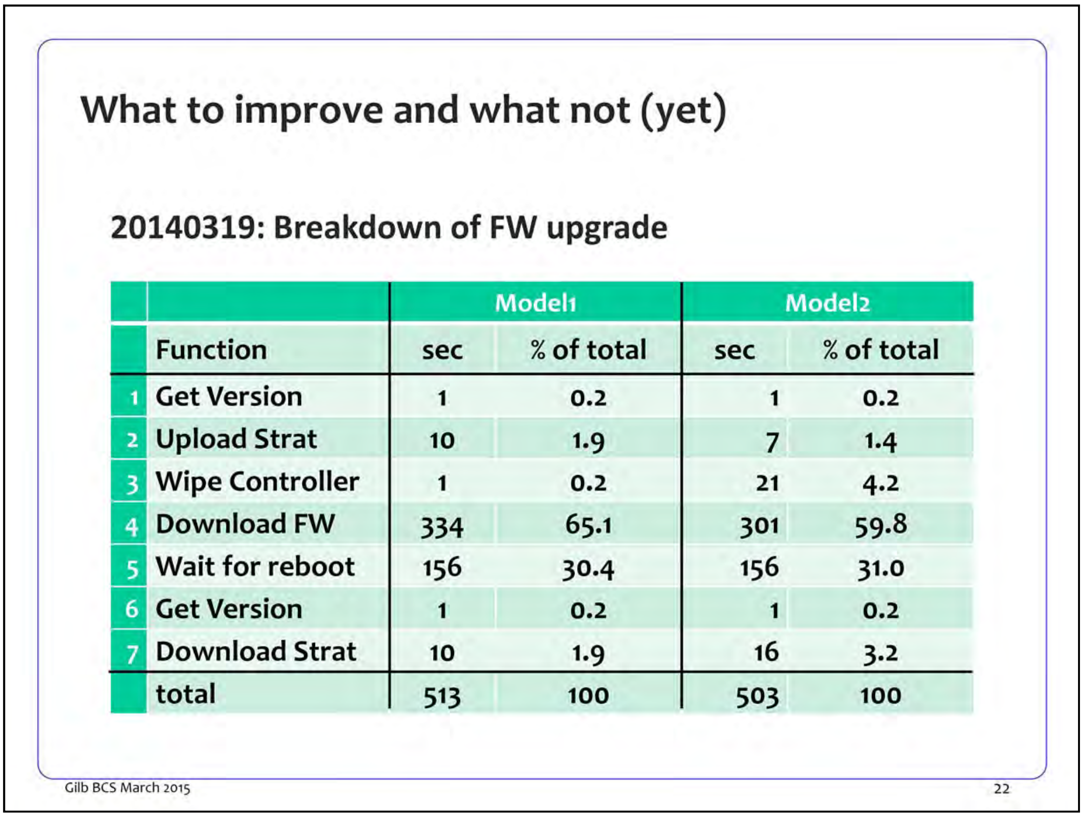
Model1: 660 secs ← Vincent, network with 16 Model1s & 16 Model2s

Model2: 330 secs ← Vincent, network with 16 Model1s & 16 Model2s, SysTst5

Model2-64: 903 secs ← Vincent, network with 64 Model2s

This went through several versions.

Vincent added some actual measurements.



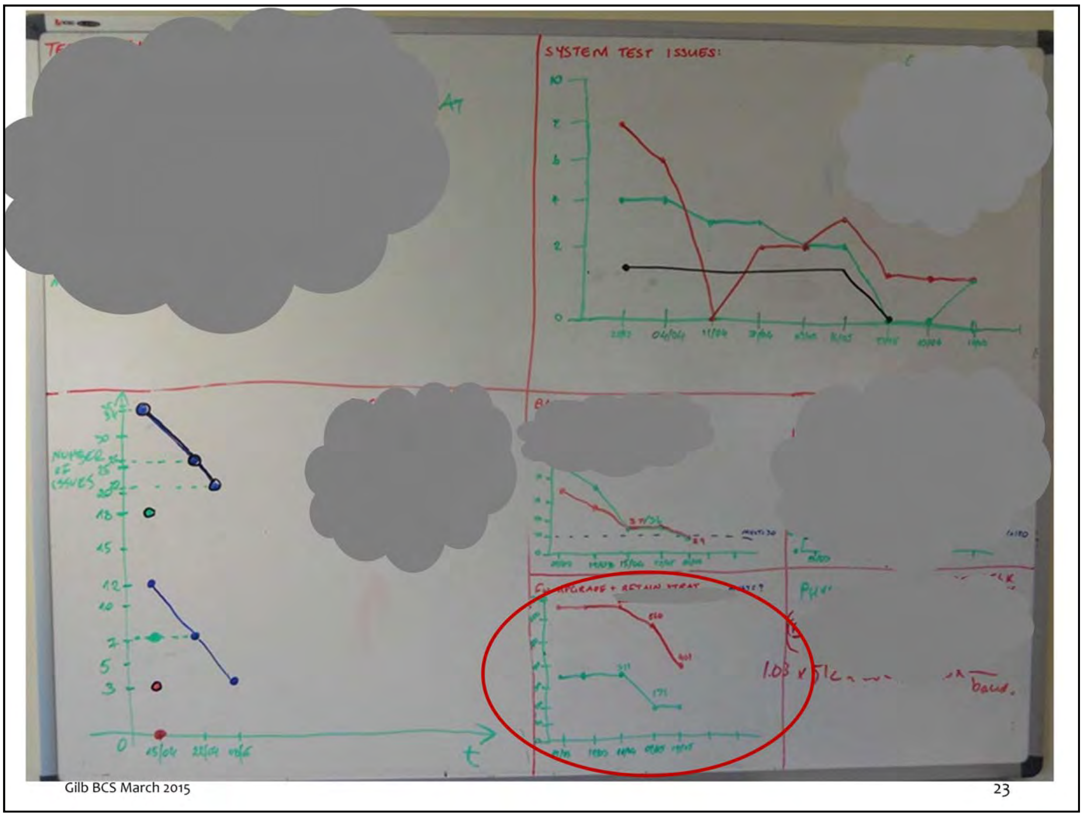
What to improve and what not (yet)

20140319: Breakdown of FW upgrade

		Model1		Model2	
	Function	sec	% of total	sec	% of total
1	Get Version	1	0.2	1	0.2
2	Upload Strat	10	1.9	7	1.4
3	Wipe Controller	1	0.2	21	4.2
4	Download FW	334	65.1	301	59.8
5	Wait for reboot	156	30.4	156	31.0
6	Get Version	1	0.2	1	0.2
7	Download Strat	10	1.9	16	3.2
	total	513	100	503	100

To see *where* it would be useful to look for improvement, Vincent measured where the time for download was spent.

We don’t have to focus on those elements which cost only 1 sec, as long as there were other elements where an improvement could create much more time-saving.

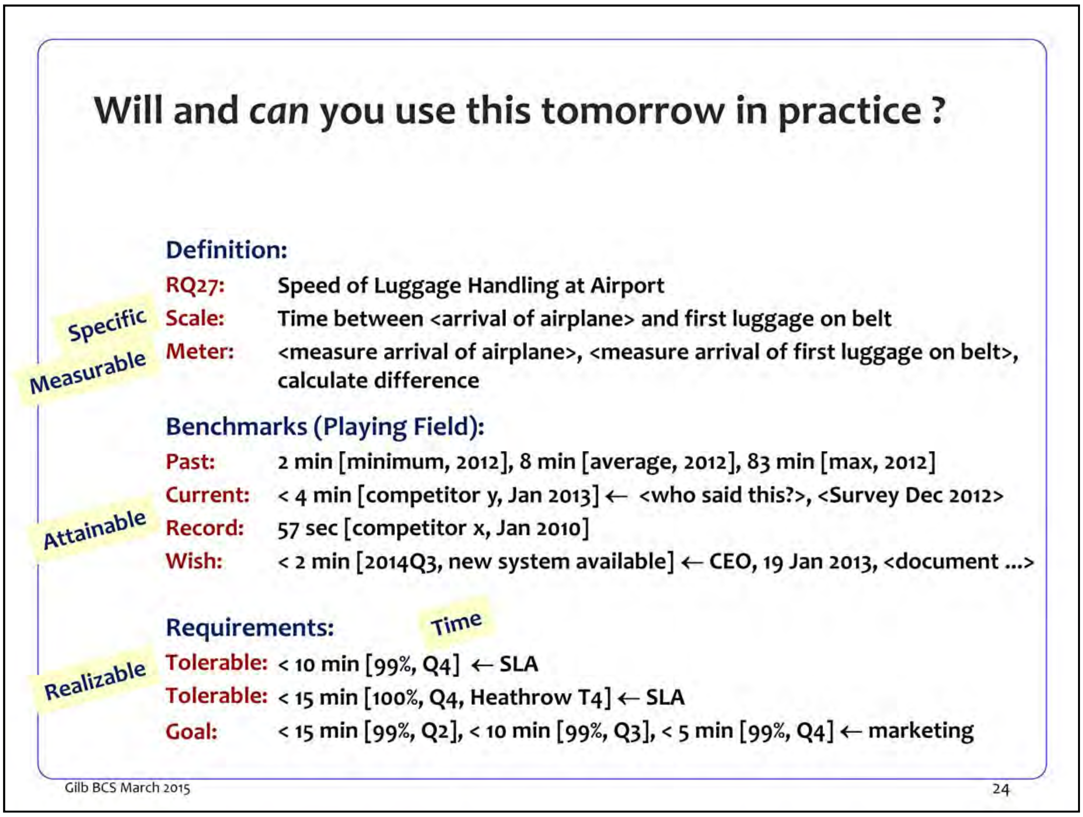


This was the white-board showing improvements after subsequent evolutionary steps, not only in the download time, but also in the removal of defects (sorry, here I cannot show you everything).

I asked “What is the physical limit of the download process?”

I got a big question-mark on the faces.

In order to transmit a certain number of bits at a certain baud-rate (bits per second), we need (number of bits)/(baud-rate) seconds, which is the physical limit. Trying to improve the system beyond this limit costs effort without any improvement. So if you don’t know this limit, you don’t know when to stop.



Will and *can* you use this tomorrow in practice ?

Definition:

Specific
Measurable

RQ27: Speed of Luggage Handling at Airport
Scale: Time between <arrival of airplane> and first luggage on belt
Meter: <measure arrival of airplane>, <measure arrival of first luggage on belt>, calculate difference

Benchmarks (Playing Field):

Attainable

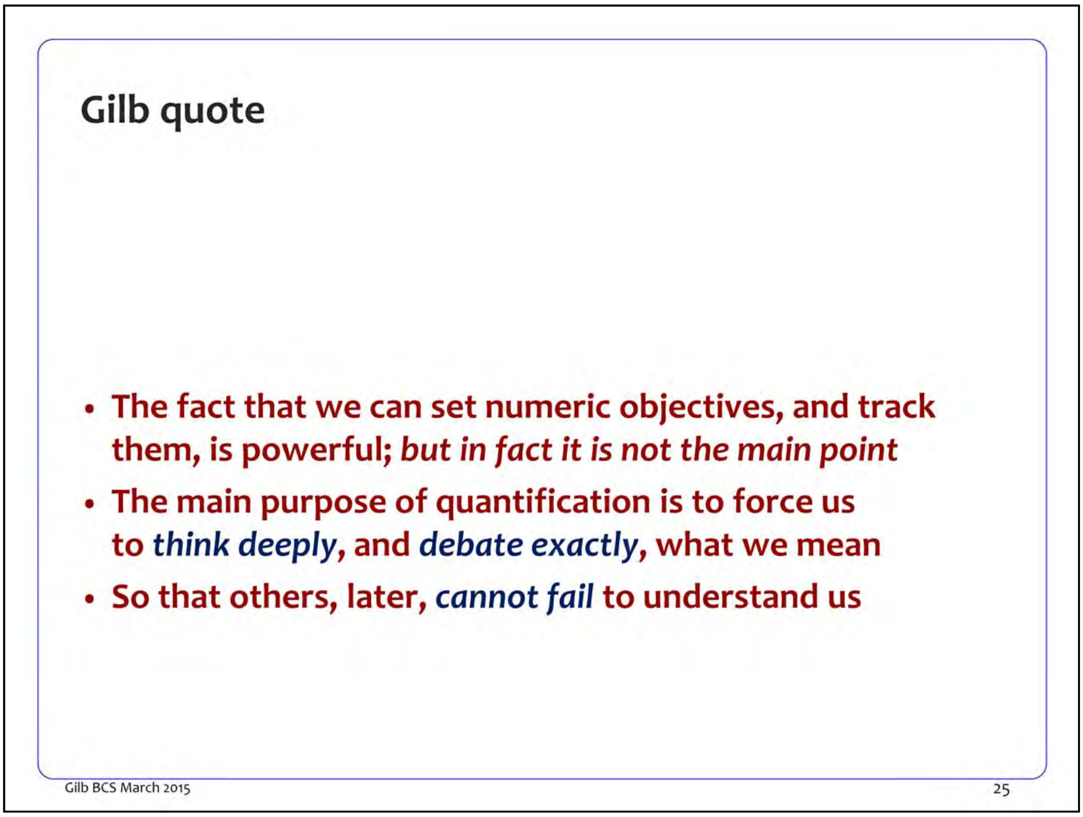
Past: 2 min [minimum, 2012], 8 min [average, 2012], 83 min [max, 2012]
Current: < 4 min [competitor y, Jan 2013] ← <who said this?>, <Survey Dec 2012>
Record: 57 sec [competitor x, Jan 2010]
Wish: < 2 min [2014Q3, new system available] ← CEO, 19 Jan 2013, <document ...>

Requirements: Time

Realizable

Tolerable: < 10 min [99%, Q4] ← SLA
Tolerable: < 15 min [100%, Q4, Heathrow T4] ← SLA
Goal: < 15 min [99%, Q2], < 10 min [99%, Q3], < 5 min [99%, Q4] ← marketing

What are you going to do next week, when you're back at work?
Will you immediately start using any of these ideas?



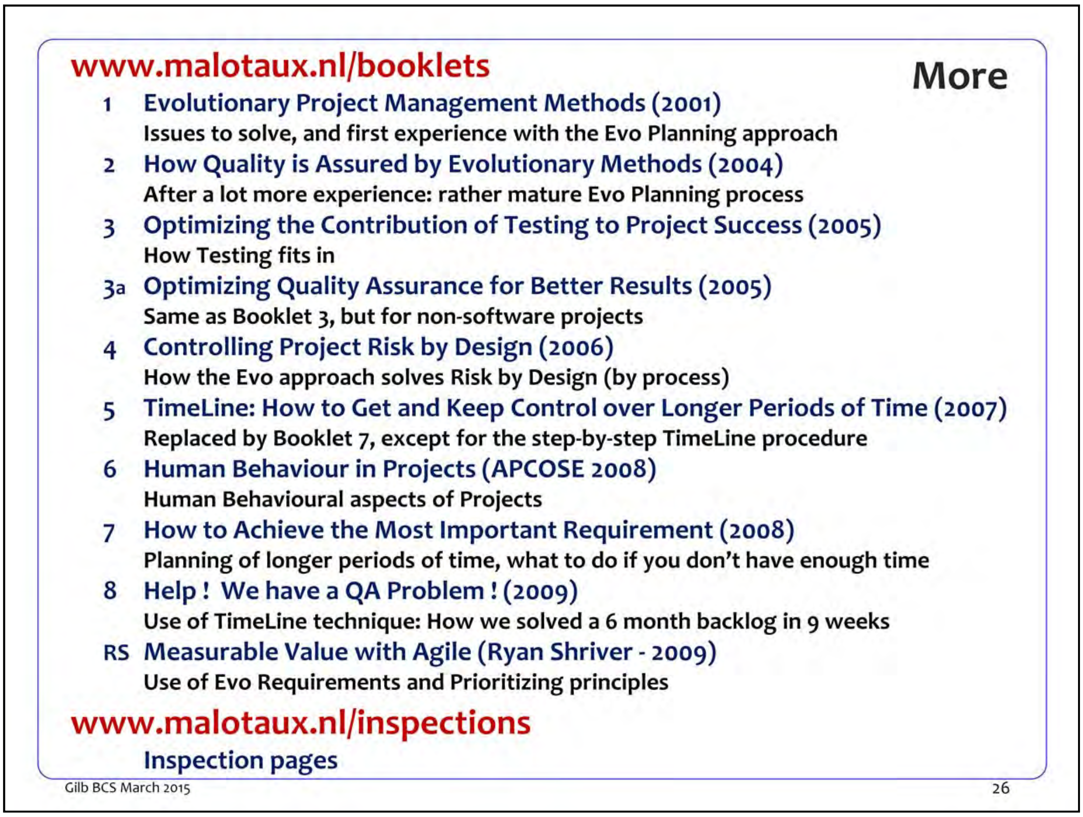
Gilb quote

- The fact that we can set numeric objectives, and track them, is powerful; *but in fact it is not the main point*
- The main purpose of quantification is to force us to *think deeply*, and *debate exactly*, what we mean
- So that others, later, *cannot fail* to understand us

Gilb BCS March 2015

25

I think this Gilb-quote is important to remember.



More

www.malotaux.nl/booklets

1 **Evolutionary Project Management Methods (2001)**
Issues to solve, and first experience with the Evo Planning approach

2 **How Quality is Assured by Evolutionary Methods (2004)**
After a lot more experience: rather mature Evo Planning process

3 **Optimizing the Contribution of Testing to Project Success (2005)**
How Testing fits in

3a **Optimizing Quality Assurance for Better Results (2005)**
Same as Booklet 3, but for non-software projects

4 **Controlling Project Risk by Design (2006)**
How the Evo approach solves Risk by Design (by process)

5 **TimeLine: How to Get and Keep Control over Longer Periods of Time (2007)**
Replaced by Booklet 7, except for the step-by-step TimeLine procedure

6 **Human Behaviour in Projects (APCOSE 2008)**
Human Behavioural aspects of Projects

7 **How to Achieve the Most Important Requirement (2008)**
Planning of longer periods of time, what to do if you don't have enough time

8 **Help ! We have a QA Problem ! (2009)**
Use of TimeLine technique: How we solved a 6 month backlog in 9 weeks

RS **Measurable Value with Agile (Ryan Shriver - 2009)**
Use of Evo Requirements and Prioritizing principles

www.malotaux.nl/inspections

Inspection pages

How to get the message across

Niels Malotaux

N R Malotaux
Consultancy

+31-655 753 604 niels@malotaux.nl www.malotaux.nl