# "HOW TO QUALITY CONTROL AND MEASURE QUALITY OF DESIGN AND ARCHITECTURE USING PLANGUAGE AND SPEC QC"

Draft 0.1, i.e. rough cut

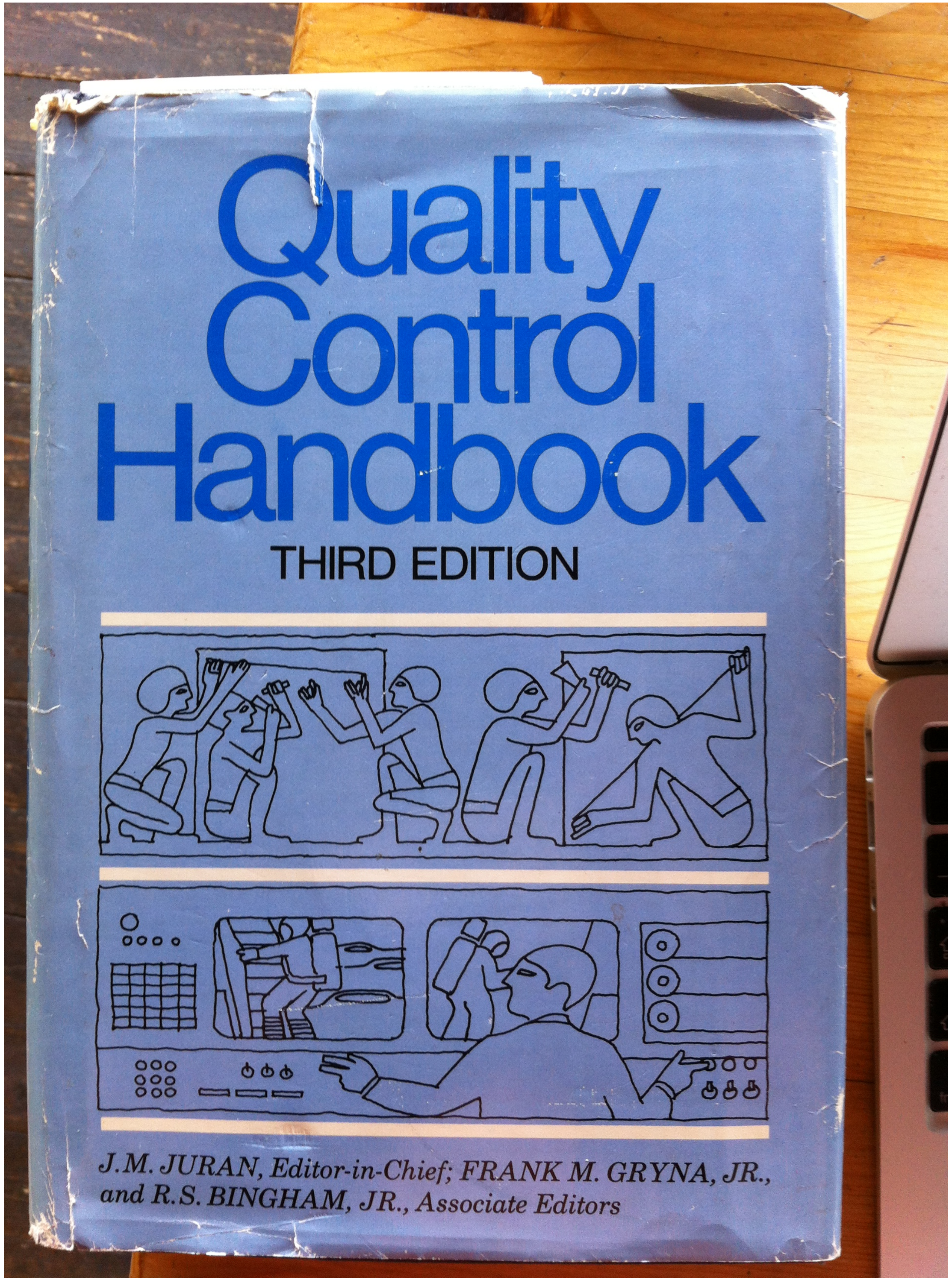
for

GilbFest Friday 26 June 2015

12:10 to 13:00

(20 minutes lecture, + 30 discussion)

QC
to a
Standard



Source Eric Simmons, erik.simmons@intel.com 25 Oct 2011
http://selab.fbk.eu/re11_download/industry/Terzakis.pdf

# A Recent Example

Application of Specification Quality Control by a SW team resulted in the following defect density reduction in requirements over several months:

| Rev. | # of Defects | # of Pages | Defects/ Page (DPP) | % Change in DPP |
|---|---|---|---|---|
| 0.3 | 312 | 31 | 10.06 | |
| 0.5 | 209 | 44 | 4.75 | -53% |
| 0.6 | 247 | 60 | 4.12 | -13% |
| 0.7 | 114 | 33 | 3.45 | -16% |
| 0.8 | 45 | 38 | 1.18 | -66% |
| 1.0 | 10 | 45 | 0.22 | -81% |
| Overall % change in DPP revision 0.3 to 1.0: | | | | -98% |

Downstream benefits:

•Scope delivered at the Alpha milestone increased 300%, released scope up 233%

•SW defects reduced by ~50%

•Defects that did occur were resolved in far less time on average

• teams typically exit with densities ranging from 5 majors per page (600 words) to 1 defect in a couple of pages.

# We are first going to look at QC of *design specifications* themselves

Based on Competitive Engineering
Design Chapter
https://www.dropbox.com/s/usfylrnek9dadsq/185%20Ch007%20Design%20ideas%20and%20Design%20Engineering.pdf?dl=0

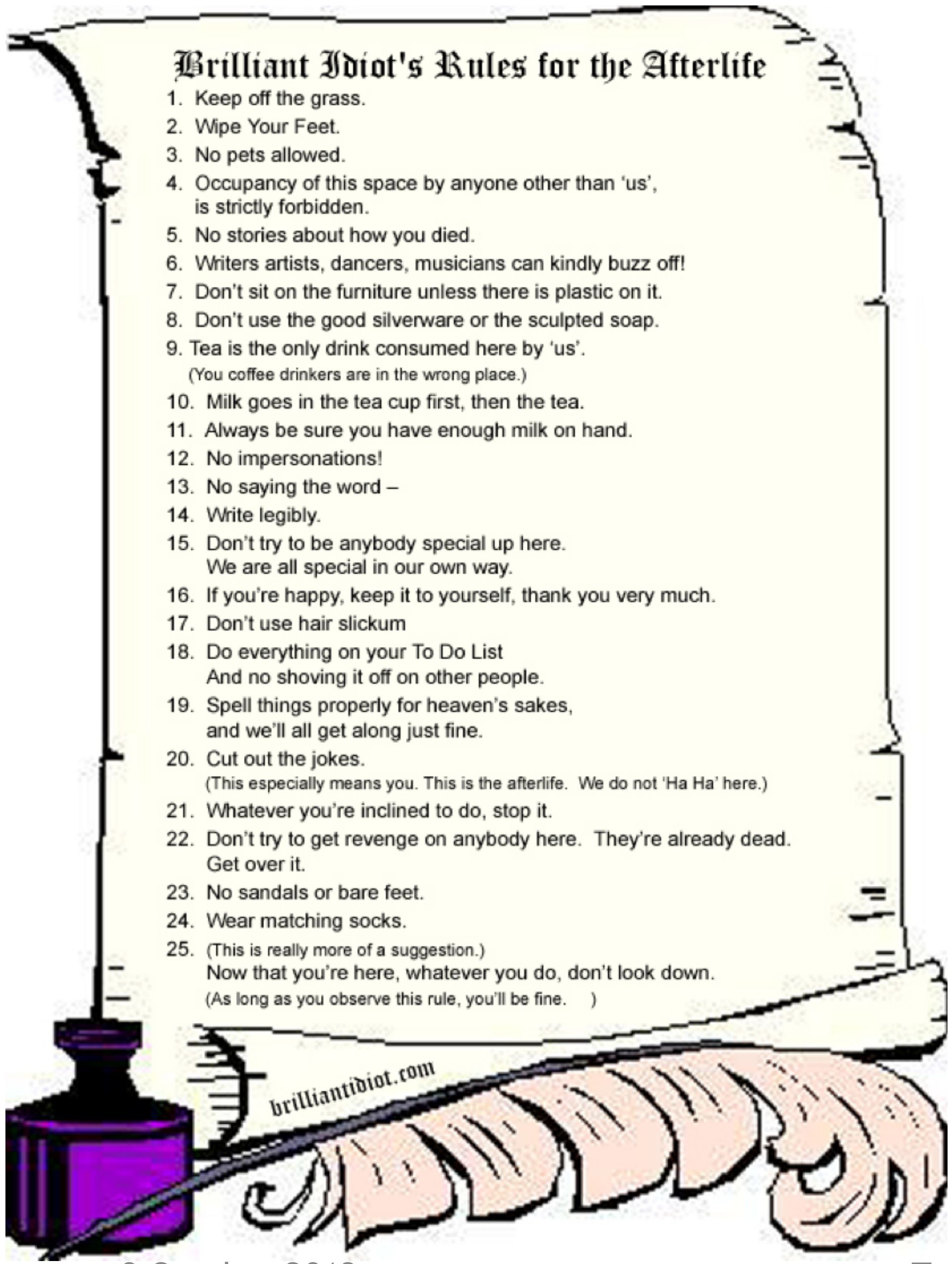
or whole CE book
https://www.dropbox.com/sh/jneaayejpf2hmdm/AACoXqKdkUbnp_zSMi_5q0_xa?dl=0

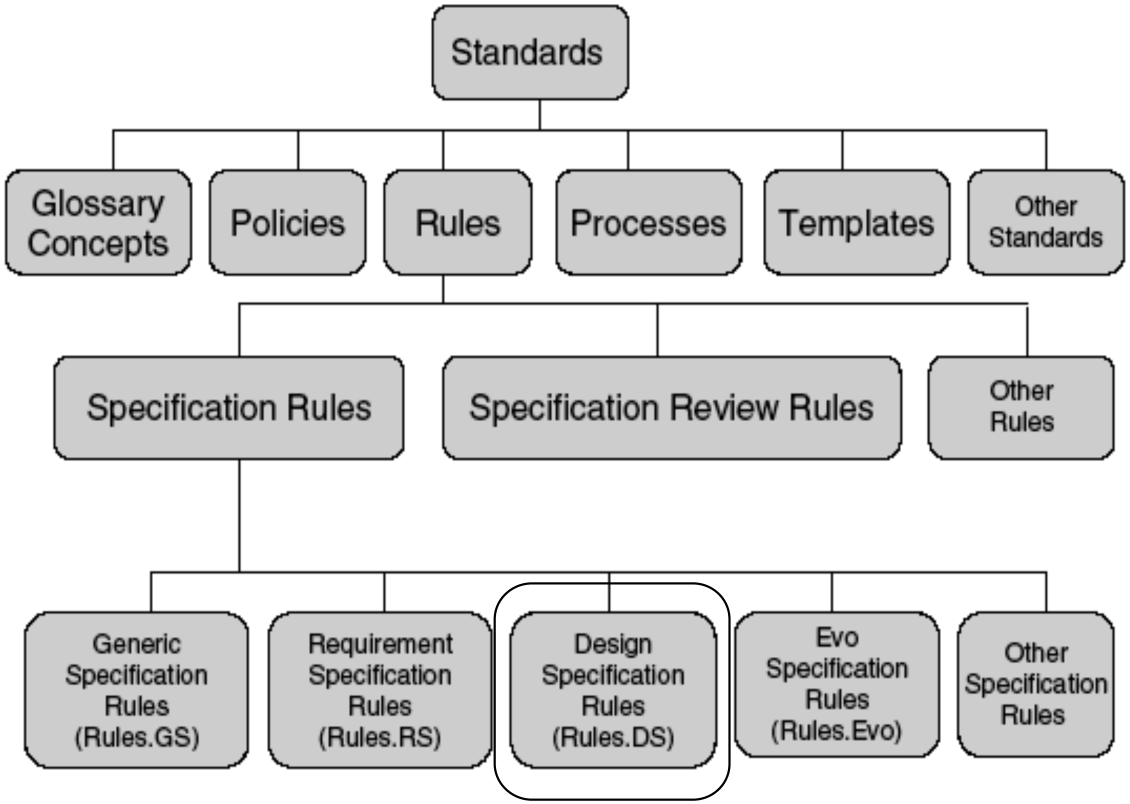
# Design Rules from Competitive Engineering, for Planguage

Version Oct 9 2013 for London Software Architect conference Keynote

By Tom Gilb

# Specification Rule Types: useful for Architecture Processes and Specification 3




See next slide
For detailed example

# Architecture Specification Rules

from CE Book Ch. 7


## 7.4 Rules: Design Specification

(edited down for simplicity)

R1: Design Separation: Only design ideas that are intentionally ‘constraints’ (Type: Design Constraint) are specified in the requirements. Any other design ideas are specified separately (Type: Design Idea).

R2: Detail: A design specification should be specified in enough detail so that we know precisely what is expected, and do not, and cannot, inadvertently assume or include design elements, which are not actually intended.

R3: Explode: Any design idea (Type: Complex Design Idea), whose impact on attributes can be better controlled by detailing it, should be broken down into a list of the tag names of its elementary and/or complex sub-design ideas.

R4: Dependencies: Any known dependencies for successful implementation of a design idea need to be specified explicitly.

R5: Impacts: For each design idea, specify at least one main performance attribute impacted by it. Use an impact arrow ‘->’ or the Impacts parameter.

R6: Side Effects: Document in the design specification any side effects of the design idea (on defined requirements or other specified potential design ideas) that you expect or fear. Do this using explicit parameters, such as Risks, Impacts [Side Effect] and Assumptions.

R7: Background Information: Capture the background information for any estimated or actual impact of a design idea on a performance/ cost attribute. The evidence supporting the impact, the level of, the level of credibility of any information and the source(s) for all this information should be given as far as possible.

R8: IE table: The set of design ideas specified to meet a set of requirements should be validated at an early stage by using an Impact Estimation (IE) table.

# Architecture Specification Rules

from CE Book Ch. 7


7.4 Rules: **Design Specification**

(edited down for simplicity)

## R1: Design Separation:

**Only design ideas that are intentionally 'constraints'** (Type: Design Constraint) **are specified in the requirements.**

**Any other design ideas are specified separately** (Type: Design Idea).

**Orbit Application Base:**

**Type: Primary Architecture Option**

==== Basic Information ==========

**Version**: Nov. 30 20xx 16:49, updated 2.Dec by telephone and in meeting. 14:34

**Status**: Draft (PUBLIC EXAMPLE EDIT)

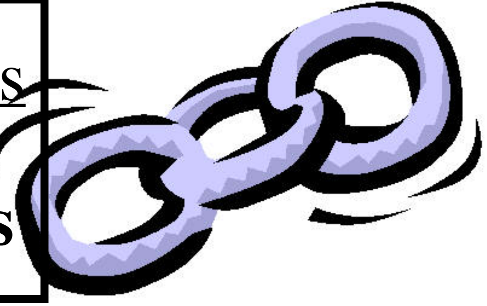
**Owner**: Brent Barclays

**Expert**: Raj Shell, London

**Authority**: for differentiating business environment characteristics, Raj Shell, Brent Barclays(for overview)

**Source**: <Source references for the information in this specification. Could include people>. Various, can be done later BB

**Gist: risk and P/L aggregation service, which also provides work flow/ adjustment and outbound and inbound feed support. Currently used by Rates Extra Business, Front Office and Middle Office, USA & UK.**



## Bad real example: Mixing Design and Requirements

OBJECTIVE (links) ARCHITECTURE

## RULE: No Design/Architecture in Requirements

- **Rationalize into a smaller number of core processing platforms.** ***This cuts technology spend*** **on duplicate platforms, and** ***creates the opportunity for*** **operational saves. Expected 60%-80% reduction in processing cost to Fixed Income Business lines.**
- **International Securities on one platform, Fixed Income and Equities (Institutional and PB).**
- **Global Processing consistency with single Operations In-Tray and associated workflow.**
- **Consistent financial processing on one Accounting engine, feeding a single sub-ledger across products.**
- **First step towards evolution of "Big Ideas" for Securities.**
- ***Improved development environment*****, leading to increased capacity to enhance functionality in future.**
- **Removes duplicative spend on two back office platforms in support of mandatory message changes, etc.**

# Architecture Specification Rules

from CE Book Ch. 7


7.4 Rules: Design Specification

## R2: Detail:

A design specification should be specified in **enough detail** so that we know precisely what is expected,

and do not, and cannot,

inadvertently assume or include design elements, which are not actually intended.

**This is a BAD example, but a real one. Too many undefined ideas. Too many MAJOR DEFECTS. Need rewrite!**

**D1: ETL Layer. Rules based highly configurable implementation of the ETL Pattern, which allows the data to be onboarded more quickly. Load and persist new data very quickly. With minimal development required**

# Architecture Specification Rules

from CE Book Ch. 7


7.4 Rules: Design Specification

## R3: Explode:

**Any design idea**

(Type: Complex Design Idea),

**whose impact on attributes *can be better controlled by detailing it*, should be <u>broken down into a list</u> of the tag names of its elementary and/or complex sub-design ideas.**

**Description: <Describe the design idea in sufficient detail to support the estimated impacts and costs given below>.**

**D1: ETL Layer. Rules based highly configurable implementation of the ETL Pattern, which allows the data to be onboarded more quickly. Load and persist new data very quickly. With minimal development required**

**D2: high performance risk and P/L aggregation processing (Cube Building).**

**D3: Orbit supports BOTH Risk and P/**

**D4: a flexible configurable workflow tool, which can be used to easily define new workflow processes**

**D5: a report definition language, which provides 90+% of the business logic contained with Orbit, allows a quick turnaround of new and enhanced reports with minimal regression testing and release procedure impact.**

**D6: Orbit GUI. Utilizes an Outlook Explorer metaphor for ease of use, and the Dxx Express Grid Control, to provide high performance Cube Interrogation Capability**

**D7: downstream feeds. A configurable event-driven data export service, which is used to generate feeds .**

# Architecture Specification Rules

from CE Book Ch. 7


7.4 Rules: Design Specification

**R4: Dependencies:**

**Any known dependencies for successful implementation of a design idea**

**need to be <u>specified explicitly.</u>**

## Dependencies:

D1: FCxx replaces Px+ in time. ? <- tsg 2.12

# Architecture Specification Rules


7.4 Rules: Design Specification

## R5: Impacts:

For each design idea, <u>specify at least one main performance attribute impacted</u> by it.

Use an impact arrow '->' or the Impacts parameter.

**D1: ETL Layer.**

**Rules based highly configurable implementation of the ETL Pattern, which allows the data to be onboarded more quickly. Load and persist new data very quickly. With minimal development required.**

**-> <u>Business-Capability-Time-To-Market, Business Scalability</u>**

# Architecture Specification Rules

from CE Book Ch. 7


7.4 Rules: **Design Specification**

**R6: Side Effects: Document**

**in the design specification any side effects**

**of the design idea**

(on defined requirements or other specified potential design ideas)

**that you expect or fear.**

**Do this using explicit parameters, such as Risks, Impacts [Side Effect] and Assumptions.**

| Goals \ Strategies | Identify Binding Compliance Requirements Strategy |
|---|---|
| Security Administration Compliance 25% ➔ 90% | 100% |
| Security Administration Performance 24 hrs ➔ 4 hrs | 75% |
| Security Administration Availability 10 hrs ➔ 24 hrs | 0% |
| Security Administration Cost 100% ➔ 60% | 50% |
| Total Percentage Impact | 225% |
| Evidence | ISAG Gap Analysis Oct-03 |
| Cost to Implement Strategy | 15 man days (US$ 5,550) |
| Credibility | 0.9 |
| Cost Adjusted Percentage Impact | 202.5% |

# Architecture Specification Rules


7.4 Rules: Design Specification

**R6: Side Effects: Document**

**in the design specification any side effects**

**of the design idea**

(on defined requirements or other specified potential design ideas)

**that you expect or fear.**

**Do this using explicit parameters, such as Risks, Impacts [Side Effect] and Assumptions.**

**==================== Priority and Risk Management ====================**

**Assumptions: <Any assumptions that have been made>.**

**A1: FCCP is assumed to be a part of Orbit. FCxx does not currently exist and is Dec 20xx 6 months into Requirements Spec. <- Picked up by TsG from dec 2 discussions AH MA JH EC.**

**Consequence: FCxx must be a part of the impact estimation and costs rating.**

**A2: Costs, the development costs will not be different. All will base on a budget of say $nn mm and 3 years. The o+**

**costs may differ slightly, like $n mm for hardware. MA AH 3 dec**

**A3:Boss X will continue to own Orbit. TSG DEC 2**

**A4: the schedule, 3 years, will constrained to a scope we can in fact deliver, OR we will be given additional budget. If not "I would have a problem" <- BB**

**A5: the cost of expanding Orbit will not be prohibitive. <- BB 2 dec**

**A6: we have made the assumption that we can integrate Oribit with PX+ in a sensible way, even in the short term <- BB**

**Dependencies: <State any dependencies for this design idea>.**

**D1: FCxx replaces Px+ in time. ? tsg 2.12**

**Risks: <Name or refer to tags of any factors, which could threaten your estimated impacts>.**

**R1. FCxx is delayed. Mitigation: continue to use Pxx <- tsg 2.12**

**R2: the technical integration of Px+ is not as easy as thought & we must redevelop Oribit**

**R3: the and or scalability and cost of coherence will not allow us to meet the delivery.**

**R4: scalability of Orbit team and infrastructure, first year especially <- BB. People, environments, etc.**

**R5: re Cross Desk reporting Requirement, major impact on technical design. Solution not currently known. Risk no solution allowing us to report all P/L**

**Issues: <Unresolved concerns or problems in the specification or the system>.**

**I1: Do we need to put the fact that we own Orbit into the objectives (Ownership). MA said, other agreed this is a huge differentiator. Dec 2.**

**I2: what are the time scales and scope now? Unclear now BB**

**I3: what will the success factors be? We don't know what we are actually being asked to do. BB 2 dec 20xx**

**I4: for the business other than flow options, there is still a lack of clarity as to what the requirements are and how they might differ from Extra and Flow Options. BB**

**I5: the degree to which this option will be seen to be useful without Intra Day. BB 2 dec**

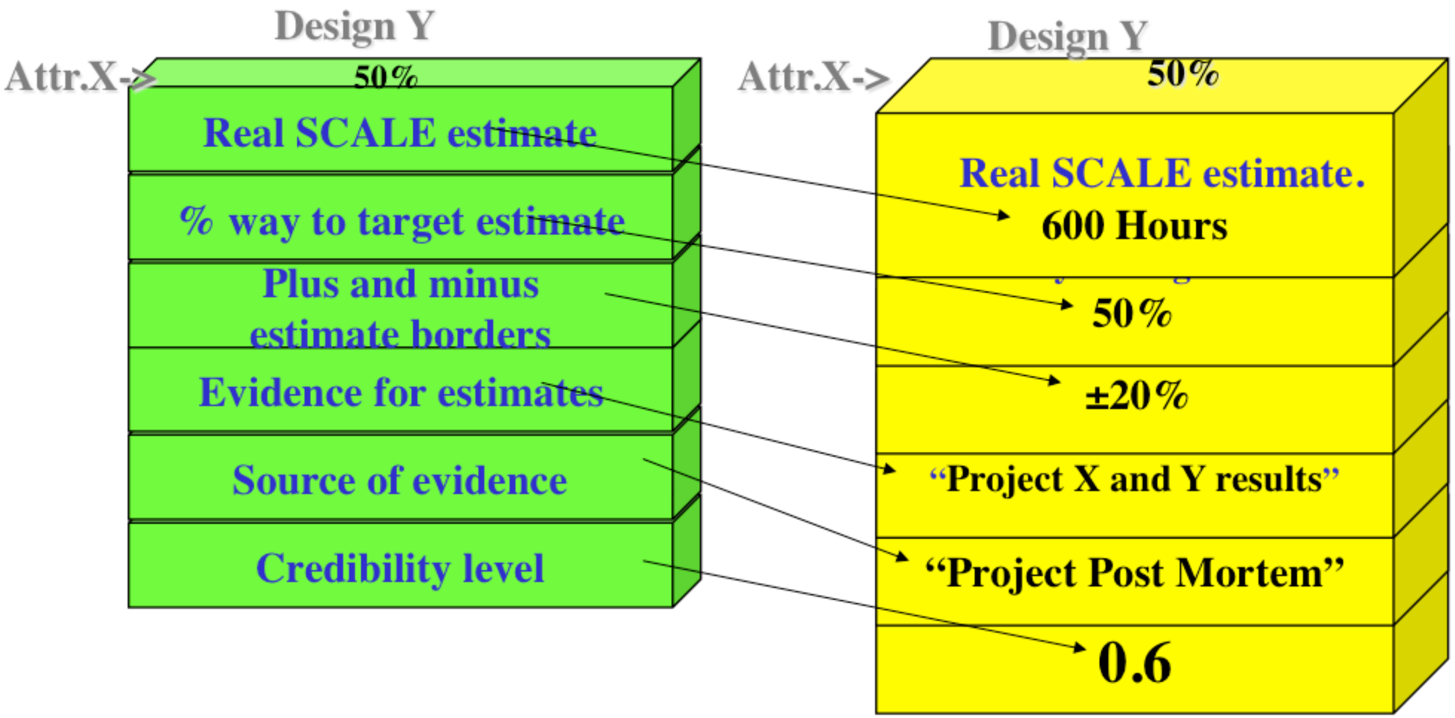
# Architecture Specification Rules

7.4 Rules: Design Specification

**R7: Background Information:**
Capture the background information for any estimated or actual impact of a design idea

on a performance/ cost attribute.

The <u>evidence</u> supporting the impact, the level of, the level of <u>credibility</u> of any information and the <u>source</u>(s) for all this information should be given as far as possible.



# Architecture Specification Rules

from CE Book Ch. 7


7.4 Rules: **Design Specification**

## R8: IE table:

**The set of design ideas specified to meet a set of requirements should be validated at an early stage by using an Impact Estimation (IE) table.**

## Acer Project: Impact Estimation Table

Strategies

Objectives

Impacts

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

See enlarged view of this slide in following slides. This is a 1-page overview

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

Orbit Application Base: (formal Cross reference Tag)

**Type**: Primary Architecture Option

============ Basic Information ==========

**Version**: Nov. 30 20xx 16:49, updated 2.Dec by telephone and in meeting. 14:34

**Status**: Draft

**Owner**: Brent Barclays

**Expert**: Raj Shell, London

**Authority**: for differentiating business environment characteristics, Raj Shell, Brent Barclays(for overview)

**Source**: <Source references for the information in this specification. Could include people>. Various, can be done later BB

**Gist**: risk and P/L aggregation service, which also provides work flow/adjustment and outbound and inbound feed support. Currently used by Rates ExtraBusiness, Front Office and Middle Office, USA & UK.

**Description**: <Describe the design idea in sufficient detail to support the estimated impacts and costs given below>.

**D1**: ETL Layer. Rules based highly configurable implementation of the ETL Pattern, which allows the data to be onboarded more quickly. Load and persist new data very quickly. With minimal development required. -> Business-Capability-Time-To-Market, Business Scalability

**D2**: high performance risk and P/L aggregation processing (Cube Building). -> Timeliness, P/L Explanation, Risk & P/L Understanding, Decision Support, Business Scalability, Responsiveness.

**D3**: Orbit supports BOTH Risk and P/L -> P/L Explanation, Risk & P/L Consistency, Risk & P/L Understanding, Decision Support.

**D4**: a flexible configurable workflow tool, which can be used to easily define new workflow processes -> Books/Records Consistency, Business Process Effectiveness, Business Capability Time to Market.

**D5:** a report definition language, which provides 90+% of the business logic contained with Orbit, allows a quick turnaround of new and enhanced reports with minimal regression testing and release procedure impact. -> P/L Explanation, Risk & P/L Understanding, Business Capability Time to Market, Business Scalability.

**D6:** Orbit GUI. Utilizes an Outlook Explorer metaphor for ease of use, and the Dxx Express Grid Control, to provide high performance Cube Interrogation Capability. -> Responsiveness, People Interchangeability, Decision Support, Risk & P/L Understanding.

**D7:** downstream feeds. A configurable event-driven data export service, which is used to generate feeds . -> Business Process Effectiveness, Business Capability Time to Market.

==================== **Priority and Risk Management** ====================

**Assumptions**: <Any assumptions that have been made>.

A1: **FCCP is assumed to be a part of Orbit.** FCxx does not currently exist and is Dec 20xx 6 months into Requirements Spec. <- Picked up by TsG from dec 2 discussions AH MA JH EC.

Consequence: FCxx must be a part of the impact estimation and costs rating.

A2: **Costs**, the development costs will not be different. All will base on a budget of say $nn mm and 3 years. The o+ costs may differ slightly, like $n mm for hardware. MA AH 3 dec

A3:Boss X will continue to own Orbit. TSG DEC 2

A4: the schedule, 3 years, will constrained to a scope we can in fact deliver, OR we will be given additional budget. If not "I would have a problem" <- BB

A5: the cost of expanding Orbit will not be prohibitive. <- BB 2 dec

A6: we have made the assumption that we can integrate Oribit with PX+ in a sensible way, even in the short term <- BB

**Dependencies**: <State any dependencies for this design idea>.

D1: FCxx replaces Px+ in time. ? tsg 2.12

**Risks**: <Name or refer to tags of any factors, which could threaten your estimated impacts>.

R1. FCxx is delayed. Mitigation: continue to use Pxx <- tsg 2.12

R2: the technical **integration** of Px+ is not as easy as thought & we must redevelop Oribit

R3: the and or scalability and cost of **coherence** will not allow us to meet the delivery.

R4: **scalability** of Orbit team and infrastructure, first year especially <- BB. People, environments, etc.

R5: re Cross Desk reporting Requirement, major impact on technical design. **Solution not currently known**. Risk no solution allowing us to report all P/L

**Issues**: <Unresolved concerns or problems in the specification or the system>.

I1: Do we need to put the fact that we own Orbit into the objectives (Ownership). MA said, other agreed this is a huge differentiator. Dec 2.

I2: what are the time scales and scope now? Unclear now BB

I3: what will the success factors be? We don't know what we are actually being asked to do. BB 2 dec 20xx

I4: for the business other than flow options, there is still a lack of clarity as to what the requirements are and how they might differ from Extra and Flow Options. BB

I5: the degree to which this option will be seen to be useful without Intra Day. BB 2 dec

| Spec Headers | Detailed Description and -> Impacted Objectives |
|---|---|
| **Orbit Application Base**: (formal Cross reference Tag)<br><br>**Type**: Primary Architecture Option<br><br>==== Basic Information ==========<br>**Version**: Nov. 30 20xx 16:49, updated 2.Dec by telephone and in meeting. 14:34<br>**Status**: Draft (PUBLIC EXAMPLE EDIT)<br>**Owner**: Brent Barclays<br>**Expert**: Raj Shell, London<br>**Authority**: for differentiating business environment characteristics, Raj Shell, Brent Barclays(for overview)<br>**Source**: <Source references for the information in this specification. Could include people>. Various, can be done later BB<br>**Gist**: risk and P/L aggregation service,<br>which also provides work flow/ adjustment and outbound and inbound feed support. Currently used by Rates Extra Business, Front Office and Middle Office, USA & UK. | **Description**: <Describe the design idea in sufficient detail to support the estimated impacts and costs given below>.<br>**D1**: ETL Layer. Rules based highly configurable implementation of the ETL Pattern, which allows the data to be onboarded more quickly. Load and persist new data very quickly. With minimal development required. -> Business-Capab<br>**D2**: high perfor<br>Timeliness, P/L<br>Business Scalab<br>**D3**: Orbit suppo<br>Consistency, Ri<br>**D4**: a flexible c<br>new workflow p<br>Effectiveness, E<br>**D5**: a report de<br>contained with<br>with minimal re<br>Explanation, Ris<br>Business Scalab<br>**D6**: Orbit GUI.<br>Dxx Express Grid Control, to provide high performance Cube interrogation Capability. -> Responsiveness, People Interchangeability, Decision Support, Risk & P/L Understanding.<br>**D7**: downstream feeds. A configurable event-driven data export service, which is used to generate feeds . -> Business Process Effectiveness, Business Capability Time to Market. |

The Detailed description is useful,

- to understand costs
- to understand impacts on your objectives
- to permit separate implementation and value delivery, incrementally

# Design Spec Enlarged 2 of 2

**==== Priority & Risk Management =========**

**Assumptions:** *<Any assumptions that have been made>.*

A1: **FCCP is assumed to** not currently exist and is Requirements Spec. <- discussions AH MA JH EC.

Consequence: FCx estimation and cos

A2: **Costs**, the developm different. All will base o and 3 years. The ops cos mm for hardware. MA AH

A3:Boss X will continue t

A4: the schedule, 3 year we can in fact deliver, O budget. If not "I would h

A5: the cost of expandin prohibitive. <- BB 2 dec

A6: we have made the assumption that we can integrate Oribit with PX+ in a sensible way, even in the short term <- BB

**Dependencies:** *<State any dependencies for this design idea>*

D1: FCxx replaces

ASSUMPTIONS:
- broadcasts critical factors for present and future re-examination
- helps risk analysis
- are an integral part of the design specifition

DEPENDENCIES:

**Risks:** *<Name or refer to tags of any factors, which could threaten your estimated impacts>.*

R1. FCxx is delaye tsg 2.12

R2: the technical thought & we mus

R3: the and or sca not allow us to me

R4: **scalability** of year especially <-

R5: re Cross Desk on technical desig Risk no solution al

Risks specification:
- shares group risk knowhow
- permits redesign to mitigate the risk
- allows relistic estimates of cost and impacts

**Issues:** *<Unresolved concerns or problems in the specification or the system>.*

I1: Do we need to put t the objectives (Owners is a huge differentiator

I2: what are the time s now BB

I3: what will the succes what we are actually b

I4: for the business oth still a lack of clarity as and how they might dif BB

I5: the degree to which this option will be seen to be useful without Intra Day. BB 2 dec

Issues:
- when answered can turn into a risk
- shares group knowledge
- makes sure we don't forget to analyze later

Part 2

# Quality Control of Impact Estimation Specifications

Based on Competitive Engineering book
Chapter on Impact Estimation
https://www.dropbox.com/s/3oad3xhlzeljjvw/261%20Ch009%20Impact%20Estimation.pdf?dl=0

or whole CE book
https://www.dropbox.com/sh/jneaayejpf2hmdm/AACoXqKdkUbnp_zSMi_5q0_xa?dl=0

# IET Rules part 1

R1: Table Format: The requirements must be specified in the left-hand column. The design ideas must be specified along the top row.

R2: Requirement: Each performance requirement (objective) and each resource requirement must be identified by its tag and by a simplified version of the chosen Baseline<->Target Pair (B<->T pair). The B<->T pair should be written under the tag.

Each B<->T pair must consist of two reference points, the chosen baseline (Past) and the planned target (Goal or Budget). Each refer- ence point must be stated as a numeric value or as a tag to a numeric value. The numeric values must be expressed using the chosen Scale for the requirement.

The baseline is stated first as it represents the 0% incremental impact point. Then usually an arrow '<->'. Then the planned target, which represents the 100% incremental impact point.

It must be possible to distinguish between multiple-level specifications for the same Goal or Budget statement. Where necessary, to be unambiguous, use a qualifier or tag the specific baseline and/or target for use in the IE table.

R3: Qualifiers: If there is one common set of qualifier [time, place and event] conditions for reaching all targets, this should be explicitly stated in the notes accompanying the IE table. If the qualifiers vary then they must be explicitly stated next to the relevant B<->T pair.

Example

By default, the entire system is implied and no specific conditions are assumed. The deadline time period must always be explicitly stated.

R4: Design Idea: Each single column must identify a design idea or set of design ideas that could be implemented as a distinct Evo step. Each design idea must be identified by its tag. Multiple tags may be specified as a set of design ideas in a single column. All tags must be supported by a design specification, which must exist in the supporting documentation and must be sufficiently detailed to allow impact estimations to the required level of accuracy. As a minimum, each design specification must be sufficiently detailed to permit financial cost to be estimated to within an 'order of magnitude.'

R5: Scale Impact: For each goal or budget, the Scale

# IET Rules part 1; 1 to 5 simplified

R1: Table Format: **The requirements must be specified in the left-hand column. The design ideas must be specified along the top row.**

R2: Requirement: **Each performance requirement (objective) and each resource requirement must be identified by its tag and by a simplified version of the chosen Baseline<->Target Pair (B<->T pair). The B<->T pair should be written under the tag.**

Format:

**Tag**

**30% <-> 75%**

R3: **Qualifiers**: If there is one common set of qualifier [time, place and event] conditions for reaching all targets, this should be **explicitly stated** in the notes accompanying the IE table.

If the qualifiers vary then they must be explicitly stated next to the relevant B<->T pair.

**The deadline time period must always be explicitly stated.**

R4: Design Idea: **Each single column must identify a design idea or set of design ideas that could be implemented as a distinct Evo step.**

**Each design idea must be identified by its tag.**

R5: **Scale Impact:** For each goal or budget, the **Scale Impact is the estimated or actual performance or cost level respectively (expressed using the relevant Scale) that is brought about by implementing the design idea(s) in each column.**

R6: **Percentage Impact**:

The Percentage Impact is a percentage (%) value derived from the Scale Impact

**An estimate of zero percent, '0%,'** means the impact of the implementation of this design idea is estimated to be equal to the specified baseline level of the objective.

**'100%' means the specified target level would probably be met exactly and on time.**

R7: **Uncertainty**: The Uncertainty (based on the evidence experience borders) of the Scale Impact estimate shall normally be specified. Percentage Uncertainty values

# IE Table Rules

## Part 2 Rules 5-10

## Full text, the 1 page of Rules for IET

R5: **Scale Impact**: For each goal or budget, the Scale Impact is the estimated or actual performance or cost level respectively (expressed using the relevant Scale) that is brought about by implementing the design idea(s) in each column.

R6: **Percentage Impact:** The Percentage Impact is a percentage (%) value derived from the Scale Impact (see Rules.IE.R2). An estimate of zero percent, '0%,' means the impact of the implementation of this design idea is estimated to be equal to the specified baseline level of the objective. '100%' means the specified target level would probably be met exactly and on time. All other percentage estimates are in relation to these two points. Note: In an IE table, it is acceptable to specify either Percentage Impacts and/or the Scale Impacts (the absolute values on the defined scale of measure). Examples: 60%, 4 minutes.

R7: **Uncertainty**: The ± Uncertainty (based on the evidence experience borders) of the Scale Impact estimate shall normally be specified. Percentage Uncertainty values are then calculated in a similar way to the Percentage Impacts. Example: 60%±20%. Usually, the uncertainty values are calculated individually for each cell. An exception to this occurs when some overall uncertainty (such as ±50%) is declared for the whole table or specified parts of it. Another more fundamental exception can be when a decision is made to defer dealing with uncertainty data.

R8: **Evidence**: Each estimate must be supported by facts that credibly show how it was derived. Numbers, dates and places are expected. If there is no evidence, a clear honest risk-identifying state- ment expressing the problem is expected (such as 'Random Guess' or 'No Evidence'). The exact source of the evidence must also be expli- citly stated. Note: Reference to a specific section of a document is permitted as evidence.

R9: **Credibility**: The evidence, together with its source, must be rated for its level of credibility on a scale of 0.0 (no credibility) to 1.0 (perfect credibility).

The relevant standard Credibility Ratings Table must be considered for use. Explanation must be given if alternative ratings are chosen.

R10: **Completeness**: All IE cells (intersections of a design idea and a requirement) must have a non-blank statement of estimated impact. This must be given as a numeric value using the relevant Scale units, or as a Percentage Impact as assessed against the defined Baseline <->Target Pair, or

# IET Rule Part 2: 6-10 simplified

R6: **Percentage Impact:**

The Percentage Impact is a percentage (%) value derived from the Scale Impact (see Rules.IE.R2).

An estimate of zero percent, '0%,' means the impact of the implementation of this design idea is estimated to be equal to the specified baseline level of the objective.

'100%' means the specified target level would probably be met exactly and on time.

All other percentage estimates are in relation to these two points.

R7: **Uncertainty**: The ±Uncertainty (based on the evidence experience borders) of the Scale Impact estimate shall normally be specified.

Percentage Uncertainty values are then calculated in a similar way to the Percentage Impacts. Example: 60% ±20%. Usually, the uncertainty values are calculated individually for each cell.

R8: **Evidence**: Each estimate must be supported by facts that credibly show how it was derived.

R9: **Credibility**: The evidence, together with its source, must be rated for its level of credibility on a scale of 0.0 (no credibility) to 1.0 (perfect credibility).

The relevant standard Credibility Ratings Table must be considered for use. Explanation must be given if alternative ratings are chosen.

R10: **Completeness**: All IE cells (intersections of a design idea and a requirement) must have a non-blank statement of estimated impact. This must be given as a numeric value using the relevant Scale units, or as a Percentage Impact as assessed against the defined Baseline <->Target Pair, or both. If there is no estimate, then a clear indication of this must be given.

R11: **Calculations**: All the appropriate IE calculations must be carried out and the arithmetic must be correct. Hint: Using an application, such as a spreadsheet, helps! The IE calculated values include:

- Percentage Impact: See Rule R6.
- Percentage Uncertainty: See Rule R7.

# Class Exercise Medical, Ward 2015
# Richard Smith's Tool

Safari File Edit View History Bookmarks Window Help 100% 12 Jun 14:15 Tom Gilb

app.needsandmeans.com

Tom Gilb & K...ents-Material appleinsider.com Google Docs TOM'S NET Services Resources » NORSKE STEDER Travel 4 TOM Social Sites NEWS ALLE ANDRE

| Requirements | Use Community Su... | Defect Preventio... | The BEST Design | Design | Sum |
|---|---|---|---|---|---|
| **Reduce Bed Days**<br>Decrease from **4** to **3** days<br>By end of November 2015<br>Average number of days per [Patient Type] per month<br>[Patient Type = Adult, child] | **0.3** days<br>**30 %** 30 % | **0.5** days<br>**50 %** 80 % | **0** days<br>**0 %** 80 % | **0** days<br>**0 %** 80 % | **80 %** |
| **Clinical Quality**<br>Decrease from **0.1** to **0.05** Infections<br>By end of June 2015<br>Average number of [Infections] per [Patient Type] per month<br>[Infections = Bloodstream, Patient Type = HIV] | **0.025** Infections<br>**50 %** 50 % | **.003** Infections<br>**6 %** 56 % | **0** Infections<br>**0 %** 56 % | **0** Infections<br>**0 %** 56 % | **56 %** |
| **Sum Of Performance:** | **80 %** 80 % | **56 %** 136 % | **0 %** 136 % | **0 %** 136 % | |
| **Skilled Effort in work Days**<br>Increase from **0** to **100** work days o...<br>By end of all<br>*No qualifiers* | **10** work d...<br>**10 %** 10 % | **50** work d...<br>**50 %** 60 % | **0** work d...<br>**0 %** 60 % | **0** work d...<br>**0 %** 60 % | **60 %** |
| **Sum Of Resources:** | **10 %** 10 % | **50 %** 60 % | **0 %** 60 % | **0 %** 60 % | |
| **Performance To Resource Ratio:** | 8.00 | 1.12 | 0.00 | 0.00 | |

# Impact Estimation Tables

Improvement

| Value Requirements Status when | Tolerable when | Goal when | Operating Model Consistency units | % of Goal |
|---|---|---|---|---|
| **P&L-Consistency&T P&L** | | | -20 | **44%** |
| 60 | 0 | 15 | -10 | 22% |
| 0 | 0 | 0 | 0.1 | 4% |
| **Speed-To-Deliver** | | | -20 | **29%** |
| 75 | 30 | 5 | -7 | 10% |
| 0 | 0 | 0 | 0.1 | 3% |
| **Operational-Control.Accurate** | | | 5 | **50%** |
| 90 | 99 | 100 | 5 | 50% |
| 0 | 0 | 0 | 0.1 | 5% |
| **Operational-Control.Consistent** | | | 1 | **50%** |
| 97 | 0 | 99 | 0.2 | 10% |
| 0 | 0 | 0 | 0.2 | 10% |
| **Operational-Control.Timely.End&Overnigh** | | | -1 | **200%** |
| 1 | 1 | 0.5 | -0.5 | 100% |
| 0 | 0 | 0 | 0.2 | 40% |
| **Operational-Control.Timely.IntradayP&L** | | | | |
| 1 | 2 | 3 | | |
| 0 | 0 | 0 | | |
| **Operational-Control.Timely.Trade-Booking** | | | -15 | **75%** |

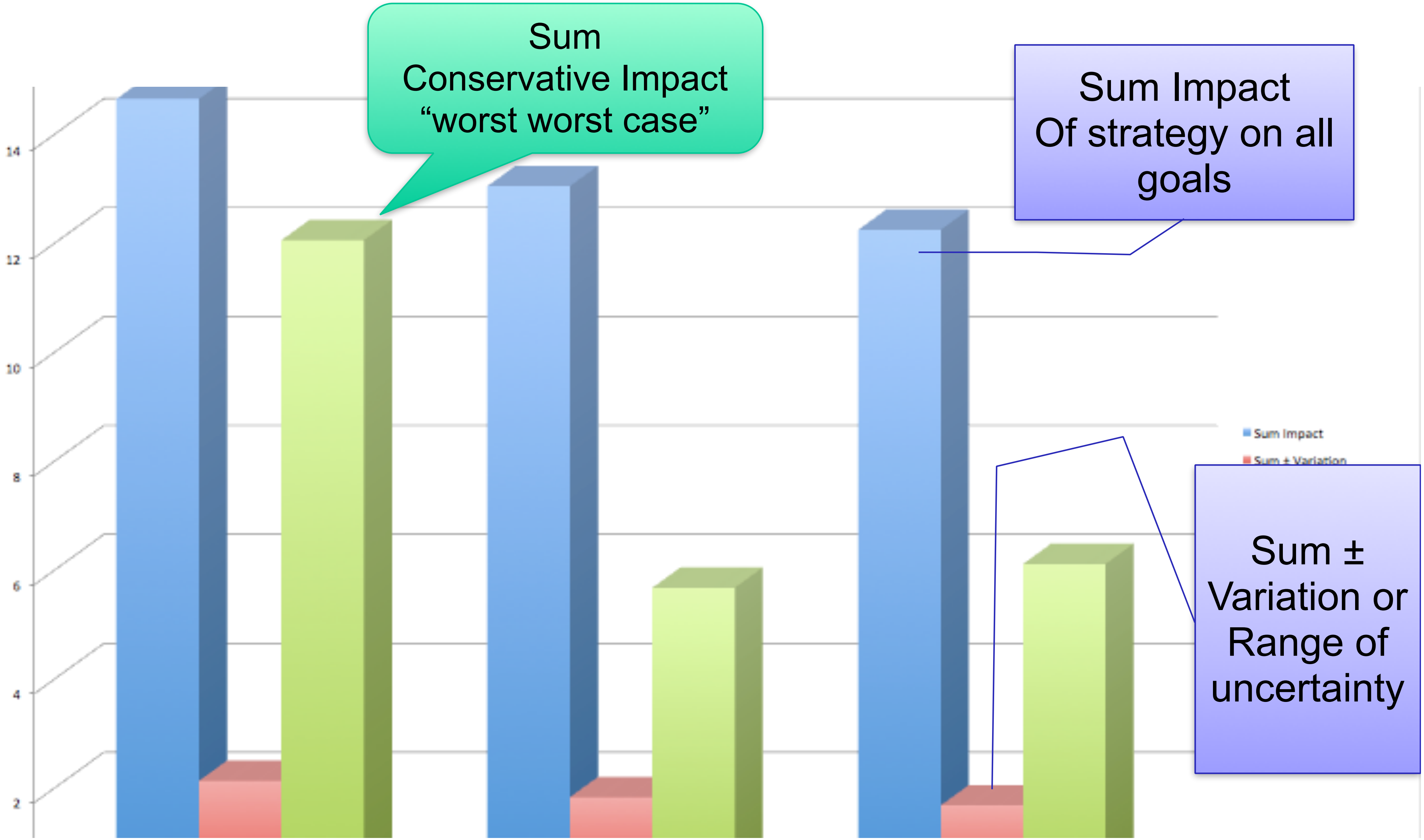
Estimate Units & %

± Uncertainty Worst Case range

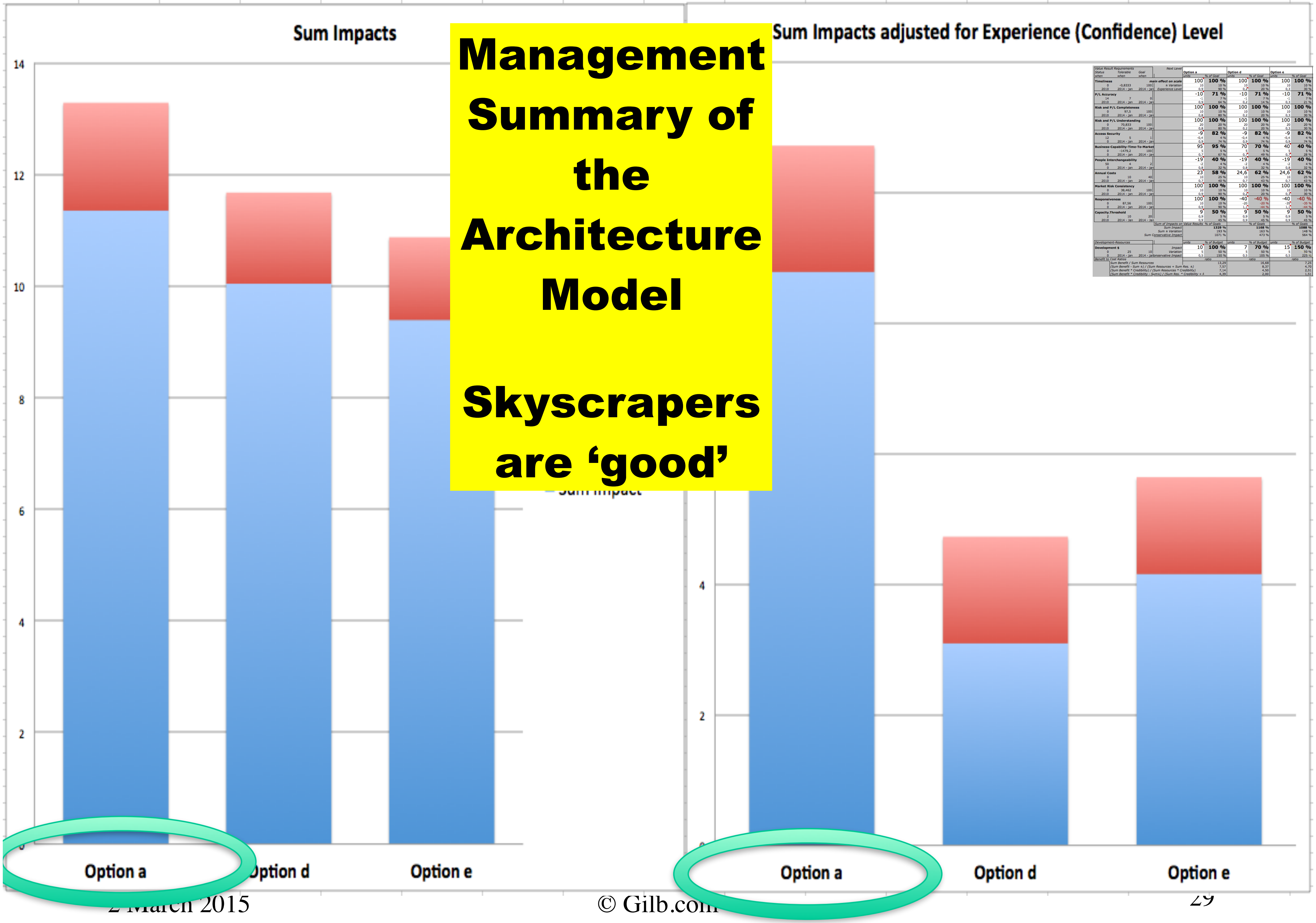
Credibility Adjustment0.0 to 1.0

Based on tool built by Kai Gilb, and his practice

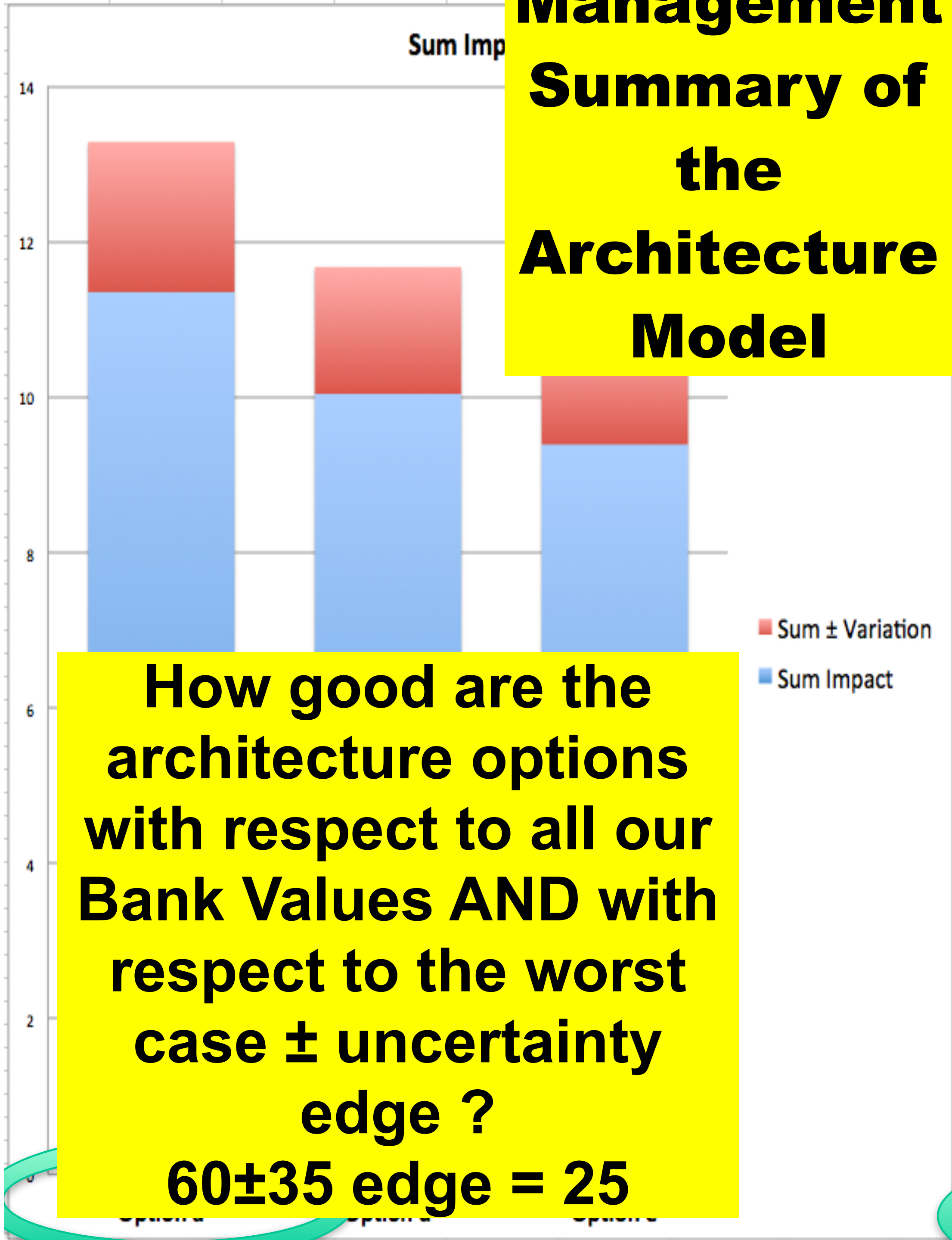
# Summary of Options wrt Risk (2010)




Based on work done by Kai Gilb

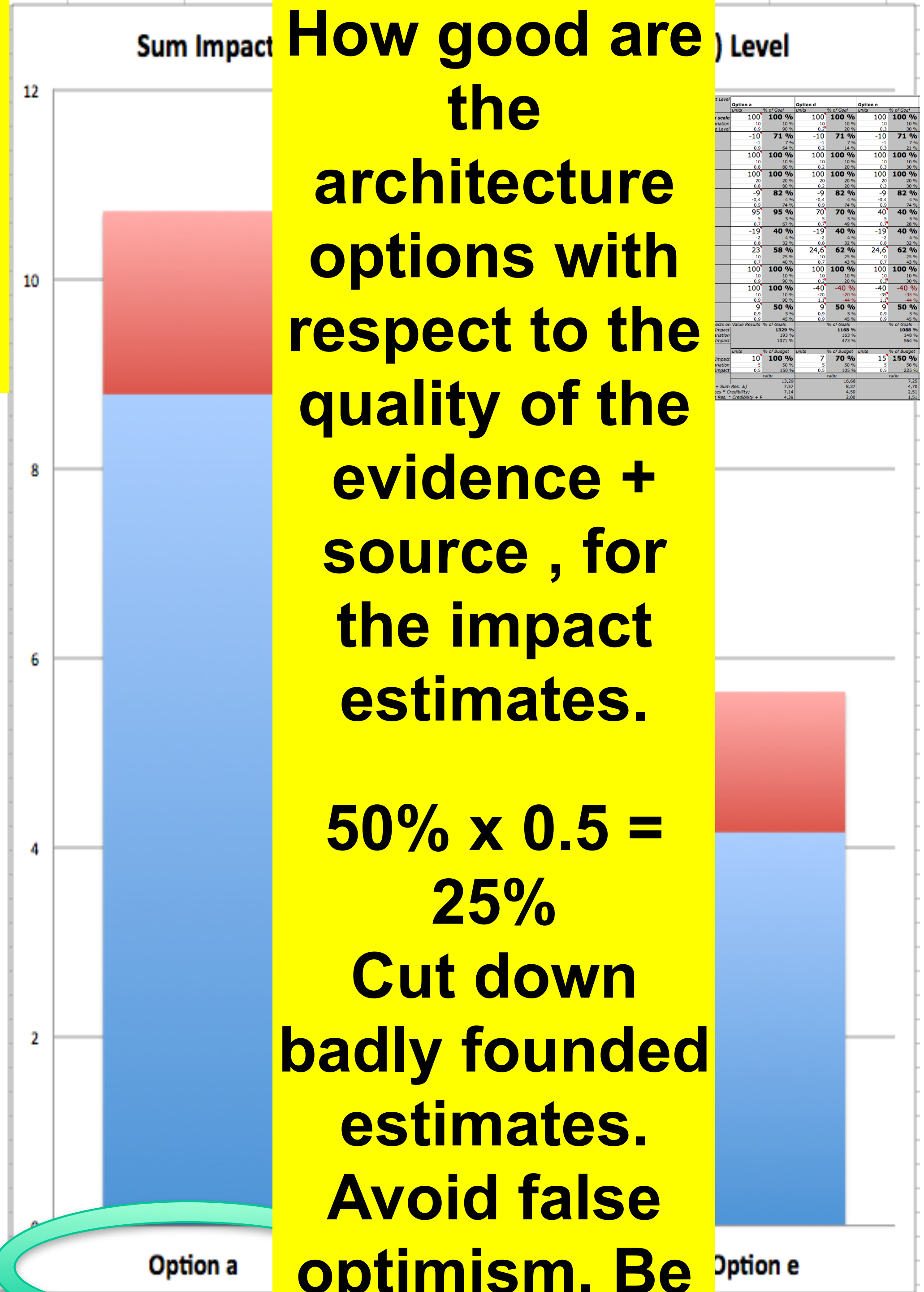

Sum Impacts
14
12
10
8
6
4
2
0
Option a
Option d
Option e
Sum Impact
Management Summary of the Architecture Model
Skyscrapers are 'good'
Sum Impacts adjusted for Experience (Confidence) Level
4
2
0
Option a
Option d
Option e

# Management Summary of the Architecture Model




How good are the architecture options with respect to all our Bank Values AND with respect to the worst case ± uncertainty edge ?

60±35 edge = 25

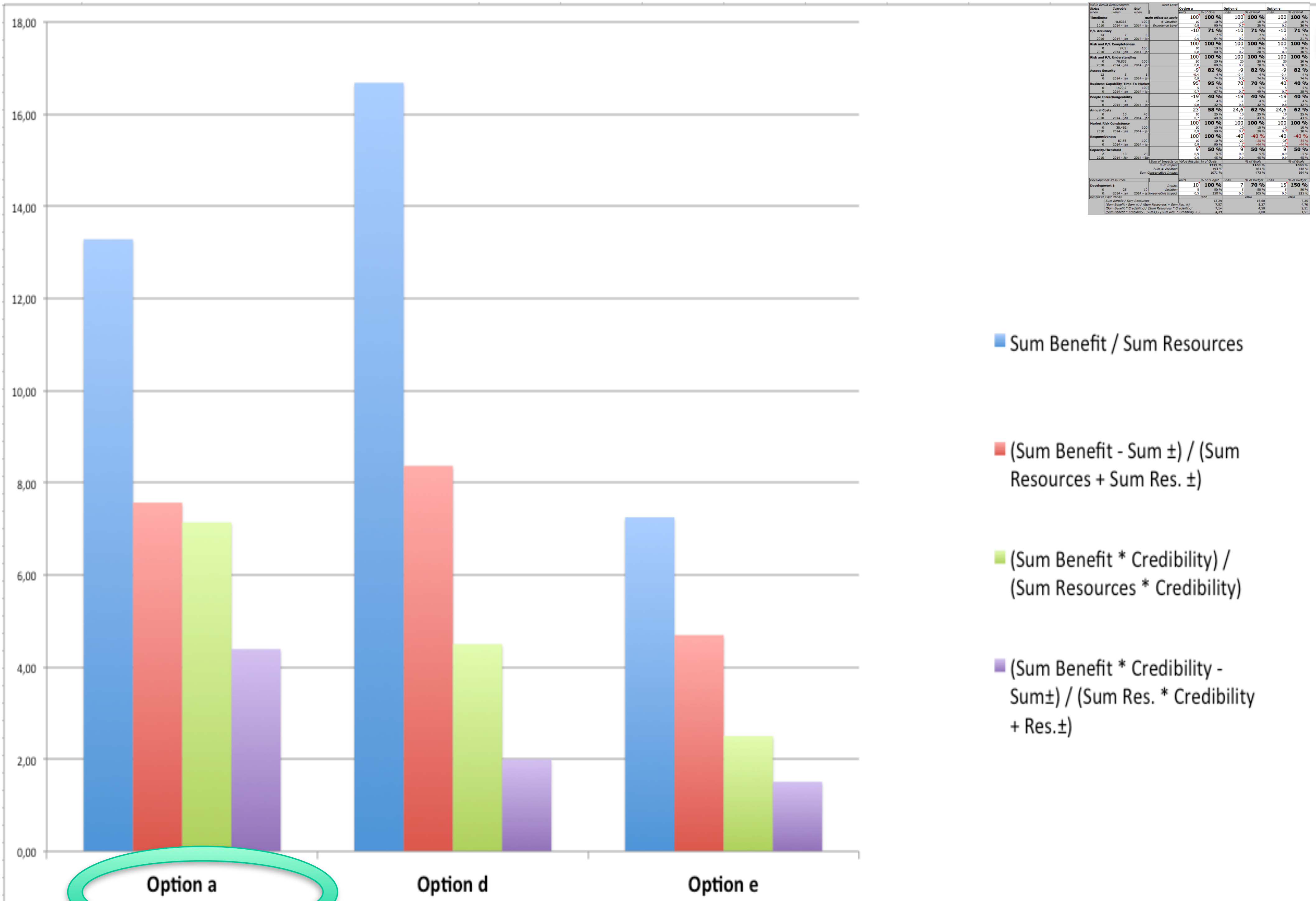



How good are the architecture options with respect to the quality of the evidence + source , for the impact estimates.

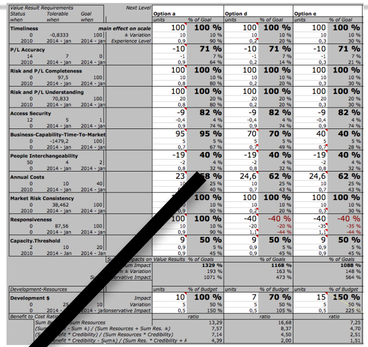
50% x 0.5 = 25%

Cut down badly founded estimates.

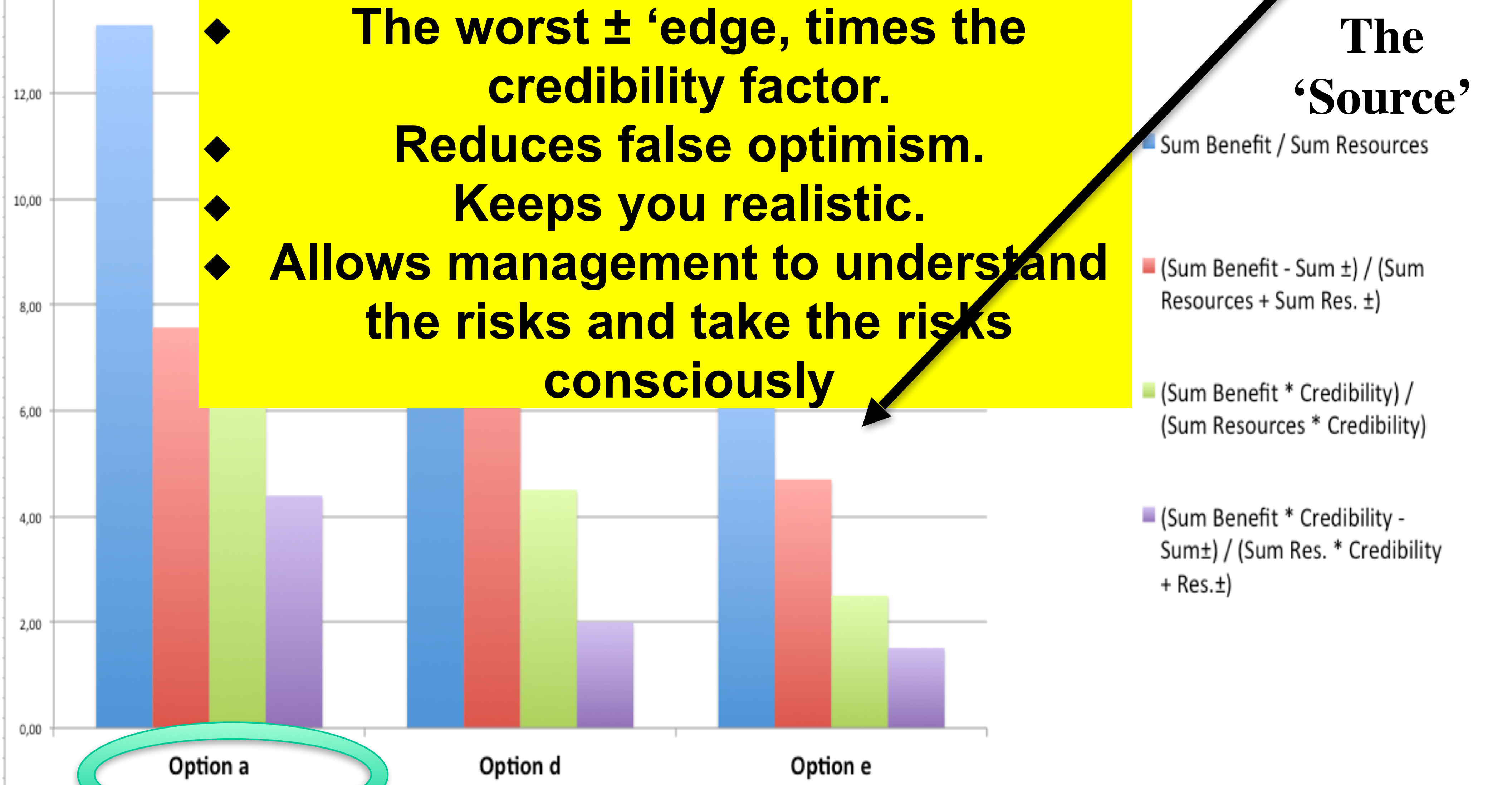
Avoid false optimism. Be realistic!


18,00
16,00
14,00
12,00
10,00
8,00
6,00
4,00
2,00
0,00
Option a
Option d
Option e
Sum Benefit / Sum Resources
(Sum Benefit - Sum ±) / (Sum Resources + Sum Res. ±)
(Sum Benefit * Credibility) / (Sum Resources * Credibility)
(Sum Benefit * Credibility - Sum±) / (Sum Res. * Credibility + Res.±)

- Combining the ‘risk’ management factors (± & Credibility Level (0.0 to 1.0) The ‘worst worst case’
  - The worst ± ‘edge, times the credibility factor.
  - Reduces false optimism.
  - Keeps you realistic.
  - Allows management to understand the risks and take the risks consciously



The ‘Source’



Part 3

# Evaluating (Reviewing) Architecture Specs, for 'RELEVANCE' to OBJECTIVES AND CONSTRAINTS

## BASIC PROCESS

Determine if

1. there is enough design to meet the goals
2. with respect to risk
3. within resource budgets

# Using Impact Estimation to get a quick initial picture of how the 7 Strategies‹#› are expected to impact the 11 Objectives and 1 cost factor.

| | Deliverables | | | | | | |
|---|---|---|---|---|---|---|---|
| | Telephony | Modularity | Tools | User Experience | GUI & Graphics | Security | Enterprise |
| **Business Objective** | | | | | | | |
| Time to Market | 10% | 10% | 15% | 0% | 0% | 0% | 5% |
| Product Range | 0% | 30% | 5% | 10% | 5% | 5% | 0% |
| Platform Technology | 10% | 0% | 0% | 5% | 0% | 10% | 5% |
| Units | 15% | 5% | 5% | 0% | 0% | 10% | 10% |
| Operator Preference | 10% | 5% | 5% | 10% | 10% | 20% | 10% |
| Commoditization | 10% | -20% | 15% | 0% | 0% | 5% | 5% |
| Duplication | 10% | 0% | 0% | 0% | 0% | 5% | 5% |
| Competitiveness | 15% | 10% | 10% | 10% | 20% | 10% | 10% |
| User Experience | 0% | 20% | 0% | 30% | 10% | 0% | 0% |
| Downstream Cost Saving | 5% | 10% | 0% | 10% | 0% | 0% | 5% |
| Other Country | 5% | 10% | 0% | 10% | 5% | 0% | 0% |
| | | | | | | | |
| Total Contribution | 90% | 80% | 55% | 85% | 50% | 65% | 55% |
| Cost (£M) | 0.49 | 1.92 | 0.81 | 1.21 | 2.68 | 0.79 | 0.60 |
| Contribution to Cost Ratio | **184** | 42 | 68 | 70 | 19 | 82 | 92 |

# DoD IE Table

| *Design Ideas ->* | *Technology Investment* | *Business Practices* | *People* | *Empowerment* | *Principles of IMA Management* | *Business Process Re-engineering* | *Sum Requirements* |
|---|---|---|---|---|---|---|---|
| Customer Service<br>? <->0 Violation of agreement | 50% | 10% | 5% | 5% | 5% | 60% | 185% |
| Availability<br>90% <-> 99.5% Up time | 50% | 5% | 5–10% | 0% | 0% | 200% | 265% |
| Usability<br>200 <-> 60 Requests by Users | 50% | 5–10% | 5–10% | 50% | 0% | 10% | 130% |
| Responsiveness<br>70% <-> ECP's on time | 50% | 10% | 90% | 25% | 5% | 50% | 180% |
| Productivity | 45% | 60% | 10% | 35% | 100% | 53% | 303% |
| 3:1 Return on Investment | 50% | 5% | 75% | 45% | 15% | 61% | 251% |
| Morale<br>72 <-> 60 per month on Sick Leave | | | | | | | |
| Data Integrity<br>88% <-> 97% Data Error % | 42% | 10% | 25% | 5% | 70% | 25% | 177% |
| Technology Adaptability<br>75% Adapt Technology | 5% | 30% | 5% | 60% | 0% | 60% | 160% |
| Requirement Adaptability<br>? <-> 2.6% Adapt to Change | 80% | 20% | 60% | 75% | 20% | 5% | 260% |
| Resource Adaptability<br>2.1M <-> ? Resource Change | 10% | 80% | 5% | 50% | 50% | 75% | 270% |
| Cost Reduction<br>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* | |

Part 4

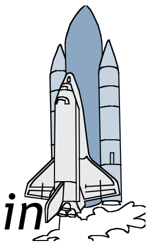
## Getting Feedback from real incremental delivery of architecture, in order to measure how well architecture really delivered values and what it costs

## Quinnans Cleanroom Process
## Confirmit Process

# Cleanroom

# In the Cleanroom Method, developed by IBM's Harlan Mi (1980) they reported:

- *"Software Engineering began to emerge in FSD" (IBM Federal Systems Division, from 1996 a part of Lockheed Martin Marietta) "some ten years ago [Ed. about 1970] in a continuing evolution that is still underway:*
- *Ten years ago general management expected the worst from software projects – cost overruns, late deliveries, unreliable and incomplete software*
- *Today [Ed. 1980!], management has learned to expect on-time, within budget, deliveries of high-quality software. A Navy helicopter ship system, called LAMPS, provides a recent example. LAMPS software was a four-year project of over 200 person-years of effort, developing over three million, and integrating over seven million words of program and data for eight different processors distributed between a helicopter and a ship in 45 incremental deliveries [Ed. Note 2%!]s. Every one of those deliveries was on time and under budget*
- *A more extended example can be found in the NASA space program,*
- *- Where in the past ten years, FSD has managed some 7,000 person-years of software development, developing and integrating over a hundred million bytes of program and data for ground and space processors in over a dozen projects.*
- *- There were few late or overrun deliveries in that decade, and none at all in the past four years."*



## In the Cleanroom Method, developed by IBM's Harlan Mi (1980) they reported:

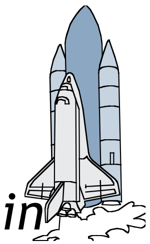
- *"Software Engineering began to emerge in FSD" (IBM Federal Systems Division, f… …bout 1…*

**in 45 incremental deliveries**

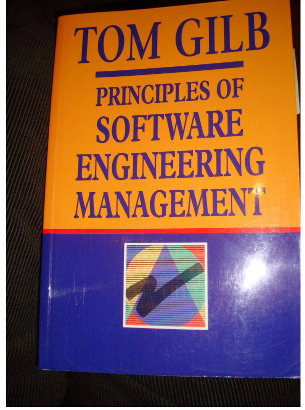
- *T… …ects – cost overruns, late deliveries, unreliable and incomplete software*
- *Today [Ed. 1980!], management has learned to expect on-time, within budget, deliveries of high-quality software. A Navy helicopter ship system, called LAMPS, provides a recent example. LAMPS software was a four-year project of over 200 person-years of effort, developing over three million, and integrating over s… …processors distri… …veries [Ed. Note … …udget*

***were few late or overrun deliveries in that decade, and none at all in the past four years***

- *A mor… …n,*
- *- Whe… …-years of softw… …million bytes … …a dozen projec…*
- *- Ther… …ne at all in the p…*



# Quinnan: IBM FSD Cleanroom
# *Dynamic Design to Cost*




**Quinnan describes the process control loop used by IBM FSD to ensure that cost targets are met.**

**'Cost management. . . yields valid cost plans linked to technical performance. Our practice carries cost management farther by introducing design-to-cost guidance. Design, development, and managerial practices are applied in an integrated way to ensure that software technical management is consistent with cost management. The method [illustrated in this book by Figure 7.10] consists of developing a design, estimating its cost, and ensuring that the design is cost-effective.' (p. 473)**

**He goes on to describe a design iteration process trying to meet cost targets by either redesign or by sacrificing 'planned capability.' When a satisfactory design at cost target is achieved for a single increment, the 'development of each increment can proceed concurrently with the program design of the others.'**

**'Design is an iterative process in which each design level is a refinement of the previous level.' (p. 474)**

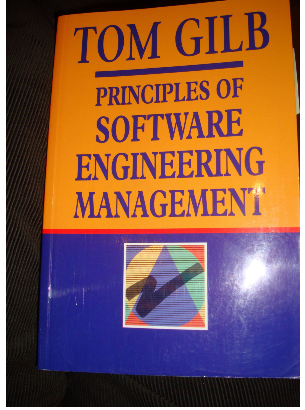
**It is clear from this that they avoid the big bang cost estimation approach. Not only do they iterate in seeking the appropriate balance between cost and design for a single increment, but they iterate through a series of increments, thus reducing the complexity of the task, and increasing the probability of learning from experience, won as each increment develops, and as the true cost of the increment becomes a fact.**

**'When the development and test of an increment are complete, an estimate to complete the remaining increments is computed.' (p. 474)**

Source: Robert E. Quinnan, 'Software Engineering Management Practices', IBM Systems Journal, Vol. 19, No. 4, 1980, pp. 466~77

This text is cut from Gilb: The Principles of Software Engineering Management, 1988

# Quinnan: IBM FSD Cleanroom
# *Dynamic Design to Cost*




**Quinnan describes the process control loop used by IBM FSD to ensure that cost targets are met.**

**'Cost management. … cost management farther by introducing design- … an integrated way to ensure that software techn… …n this book by Figure 7.10] consists of develop… …' (p. 473)**

**He goes on t… …sign or by sacrificing 'planned capability.' When a … …pment of each increment can proceed concurrent…**

**'Design is an iterati… …474)**

**of developing a design, estimating its cost, and ensuring that the design is cost-effective**

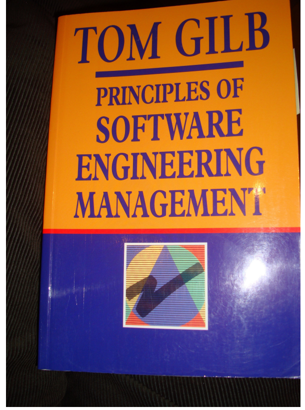
**It is clear fro… …iterate in seeking the appropriate balance between cost and design for a single increment, but they iterate through a series of increments, thus reducing the complexity of the task, and increasing the probability of learning from experience, won as each increment develops, and as the true cost of the increment becomes a fact.**

**'When the development and test of an increment are complete, an estimate to complete the remaining increments is computed.' (p. 474)**

Source: Robert E. Quinnan, 'Software Engineering Management Practices', IBM Systems Journal, Vol. 19, No. 4, 1980, pp. 466~77

This text is cut from Gilb: The Principles of Software Engineering Management, 1988

# Quinnan: IBM FSD Cleanroom
# *Dynamic Design to Cost*




**Quinnan describes the process control loop used by IBM FSD to ensure that cost targets are met.**

**'Cost management. . . yields valid cost plans linked to technical performance. Our practice carries cost management farther by introducing <u>design-to-cost guidance.</u> Design, development, and managerial practices are applied in an integrated way to ensure that software technical management is consistent with cost management. The method [illustrated in this book by Figure 7.10] consists <u>of developing a design, estimating its cost, and ensuring that the design is cost-effective.' (p. 473)</u>**

**He goes on to describe a design iteration <u>process trying to meet cost targets by either redesign or by sacrificing 'planned capability</u>.' When a satisfactory design at cost target is achieved for a single increment, the 'development of each increment can proceed concurrent**

**'<u>Design is an iterati</u> … 474)**

**It is clear fro … iterate in seeking the appropriate balance … <u>ies of increments, thus reducing the compl</u> … n as each increment develops, and <u>as th</u>**

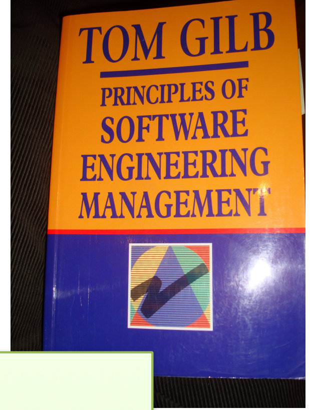
**'When the developn … <u>ing increments is computed</u>.' (p. 474)**

Source: Robert E. Qu … No. 4, 1980, pp. 466~77

This text is cut from

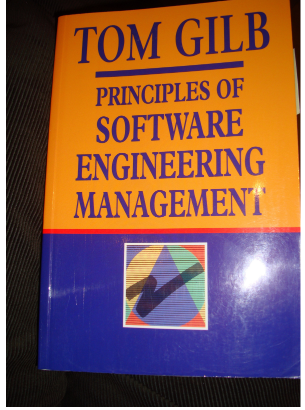
**iteration process trying to meet cost targets by <u>either</u> *redesign* or by *sacrificing* 'planned capability'**

# Quinnan: IBM FSD Cleanroom
## *Dynamic Design to Cost*




**Design is an iterative process**

# Quinnan: IBM FSD Cleanroom
## *Dynamic Design to Cost*




**Quinnan describes the process control loop used by IBM FSD to ensure that cost targets are met.**

**but they iterate through a series of increments,**
**thus *reducing the complexity of the task*,**
**and *increasing the probability of learning from experience***

# The proof is in the pudding;

**Richard Smith**

- “The proof is in the pudding;
- I have **<u>used Evo</u>**
  - *(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*** **<u>no</u> 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

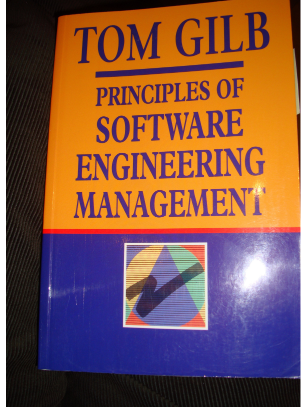
## It looks like the stakeholders liked the top level system qualities, on first try

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

# Quinnan: IBM FSD Cleanroom
## *Dynamic Design to Cost*




**Quinnan describes the process control loop used by IBM FSD to ensure that cost targets are met.**

**an estimate to complete the remaining increments is computed.**

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

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

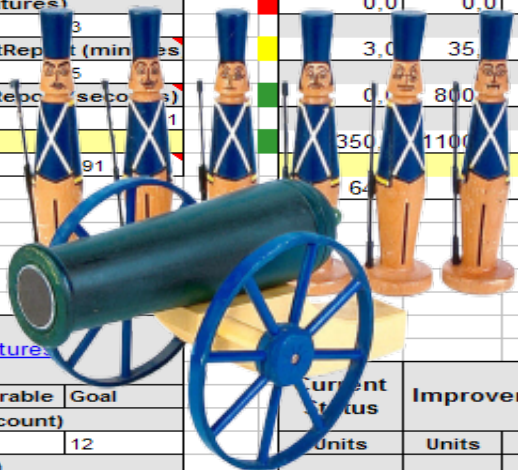
**Impact Estimation Table: Reportal codename "Hyggen"**

9

| 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[illegible]t (min[illegible]es) | | |
| 1,0 | 12,0 | 150,0 | 13 | 13 | 5 |
| | | | Usability.ResponseTime.ViewRepo[illegible] (seco[illegible]s) | | |
| 1,0 | 14,0 | 100,0 | 15 | [illegible] | 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,[illegible] | 35[illegible] | 97,2 | 38 | 3 | 2 |
| | | | Runtime.ResourceUsage.MemoryLeak | | |
| 0,[illegible] | 800 | 100,0 | 800 | 0 | 0 |
| | | | Runtime.Concurrency (number of users) | | |
| 350[illegible] | 110[illegible] | 146,7 | 150 | 500 | 1000 |
| | | | Development resources | | |
| 64[illegible] | | | 0 | | 84 |

8



3

| 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 |

3

# 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 | 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 |

Estimates

Testing

Weekly

Priority

Next week Warning metrics based

Cumulative weekly progress metric

Constraint

Target