

# ***Improving Estimations in Agile Projects: Issues and Avenues***

**Luigi Buglione & Alain Abran**



**4<sup>th</sup> Software Measurement European Forum  
9-11 May 2007, Rome, Italy**

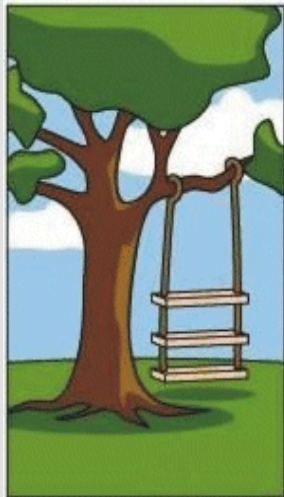
# Agenda



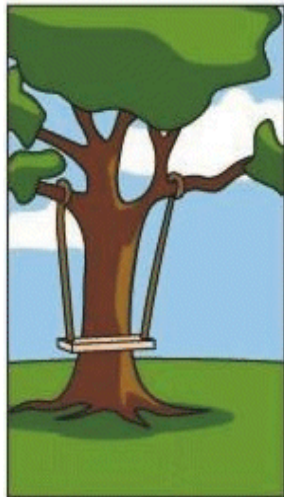
- **Introduction**
- **Agile Methodologies (AM)**
  - ✓ A Musical Analogy
  - ✓ The Agile Manifesto
  - ✓ Agile and SLC phase coverage
  - ✓ Agile and SPI Methods
- **Estimation Issues in an Agile Sw Development Project**
  - ✓ XP – Extreme Programming
  - ✓ SCRUM
  - ✓ Crystal Family
  - ✓ FDD – Feature Driven Development
  - ✓ DSDM – Dynamic Systems Development Method
  - ✓ Other studies - proposals
- **Discussion on Estimation practices in AM**
  - ✓ NFR – Non-Functional Requirements
  - ✓ Sizing Units
  - ✓ Historical Data
  - ✓ Standards
- **Conclusions**

# Introduction

## (Sw) Project Management: Common Problems (Req's)



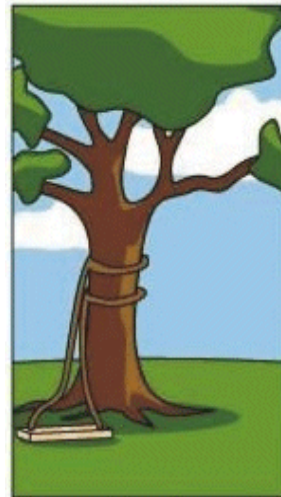
What the customer asked for



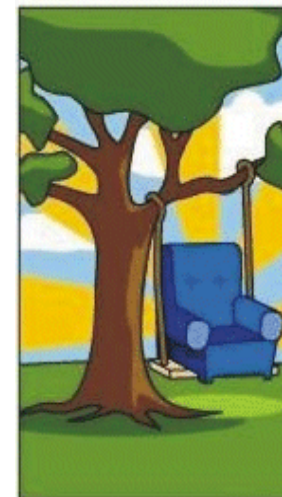
What the project manager understood



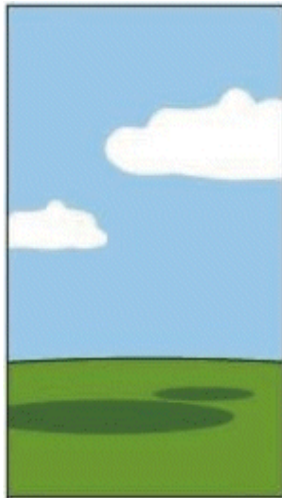
How analysts projected it



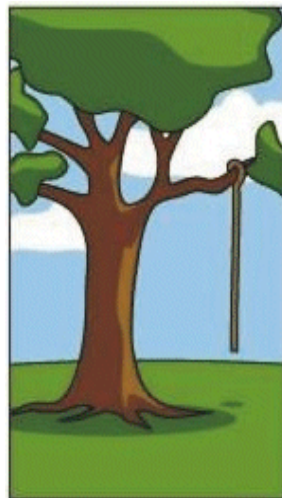
What programmers did



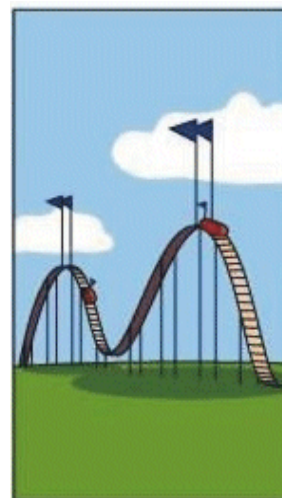
What consultant said it was needed



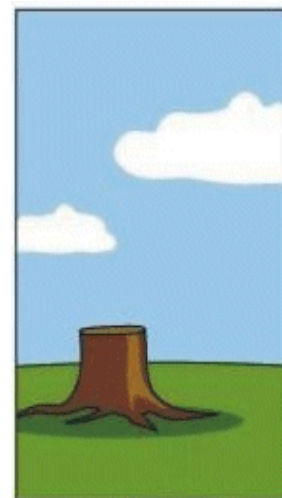
How the project was documented



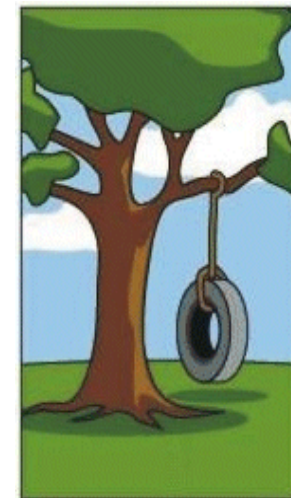
What was effectively installed



What the customer had to pay



How maintenance was run



What the customer would have needed

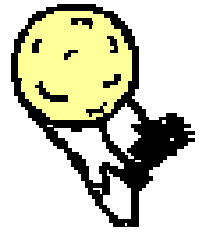
# Introduction

## (Sw) Project Management: Common Problems (Staffing)



# Introduction

## Agility: Background



- “Old” paradigms to be overcome during the '90s
  - ✓ Shorter time-to-market
  - ✓ More compressed project schedules
  - ✓ Requirements more and more instable
  - ✓ **Heavyweight** vs **lightweight** (or **agile**) methods/approaches
- “Agile” Methods (AM) as the new paradigm for Sw Development
  - ✓ A more flexible answer to manage projects
  - ✓ Common characteristics and ways to express an iterative-incremental SLC
  - ✓ Plenty of methods between mid '90 and the start of Y2K, grouped under the Agile Alliance umbrella
    - XP, SCRUM, FDD, DSDM, Crystal, ....

# Introduction

## Agility: Definition & Research Issue



- **Agility** does not mean "quicker", "faster" but...

Also Visit [Unabridged Dictionary](#) [Learner's Dictionary](#) [Word Central for Kids](#) [Collegiate Dict](#)

Merriam-Webster OnLine Search  Dictionary  Thesaurus  Spanish/English

Merriam-Webster's Online Dictionary

### agile

One entry found for **agile**.

Main Entry: **ag·ile**

Pronunciation: 'a-j&l, -'jI(-&)l

Function: *adjective*

Etymology: Middle French, from Latin *agilis*, from *agere* to drive, act -- more at [AGENT](#)

**1** : marked by ready ability to move with quick easy grace <an *agile* dancer>

**2** : having a quick resourceful and adaptable character <an *agile* mind>

- **ag·ile·ly** /-j&(I)-IE, -'jI(I)-IE/ *adverb*

[Learn more about "agile" and related topics at Britannica.com](#)

[Find more about "agile" instantly with Live Search](#)

[Pronunciation Symbols](#)

Merriam-Webster OnLine

- ▶ Home
- ▶ Visit Our Sites
- ▶ Premium Services
- ▶ Downloads
- ▶ Word of the Day
- ▶ Word Games
- ▶ Open Dictionary
- ▶ Spelling Bee Hive
- ▶ Word for the Wise
- ▶ Online Store
- ▶ Help
- ▶ About Us



**Q:**...but what about Estimation practices in Agile Methods?



# Agenda



- **Introduction**
- **Agile Methodologies (AM)**
  - ✓ A Musical Analogy
  - ✓ The Agile Manifesto
  - ✓ Agile and SLC phase coverage
  - ✓ Agile and SPI Methods
- **Estimation Issues in an Agile Sw Development Project**
  - ✓ XP - Extreme Programming
  - ✓ SCRUM
  - ✓ Crystal Family
  - ✓ FDD – Feature Driven Development
  - ✓ DSDM – Dynamic Systems Development Method
  - ✓ Other studies - proposals
- **Discussion on Estimation practices in AM**
  - ✓ NFR – Non-Functional Requirements
  - ✓ Sizing Units
  - ✓ Historical Data
  - ✓ Standards
- **Conclusions**

# Agile Methodologies (AM)

## A Musical Analogy

	<b>Symphonic Orchestra</b>	<b>Jazz Group</b>
	<b>Heavyweight</b>	<b>Lightweight-Agile</b>
		
<b>Project</b>	<ul style="list-style-type: none"> <li>• Each music session is a replay of a "production series"</li> </ul>	<ul style="list-style-type: none"> <li>• Each music session is a sort of "prototype"</li> </ul>
<b>Process</b>	<ul style="list-style-type: none"> <li>• Structured process in a rigid manner (<i>following the music partition 'as is' in the music book</i>)</li> </ul>	<ul style="list-style-type: none"> <li>• Dynamic and adaptive process (<i>no music book</i>)</li> </ul>
<b>Team</b>	<ul style="list-style-type: none"> <li>• Many people well synchronized with actions</li> </ul>	<ul style="list-style-type: none"> <li>• Tight-knit team with few (<i>and skillful</i>) musicians</li> </ul>
<b>Output</b>	<ul style="list-style-type: none"> <li>• Stable and predictable outcomes</li> </ul>	<ul style="list-style-type: none"> <li>• Improvised and variable, but staying with established beats</li> </ul>



# Agile Methodologies (AM)

## The Agile Manifesto (2001)

*"We are uncovering better ways of developing software  
by doing it and helping others do it.  
Through this work we have come to value:*

***Individuals and interactions*** over processes and tools

***Working software*** over comprehensive documentation

***Customer collaboration*** over contract negotiation

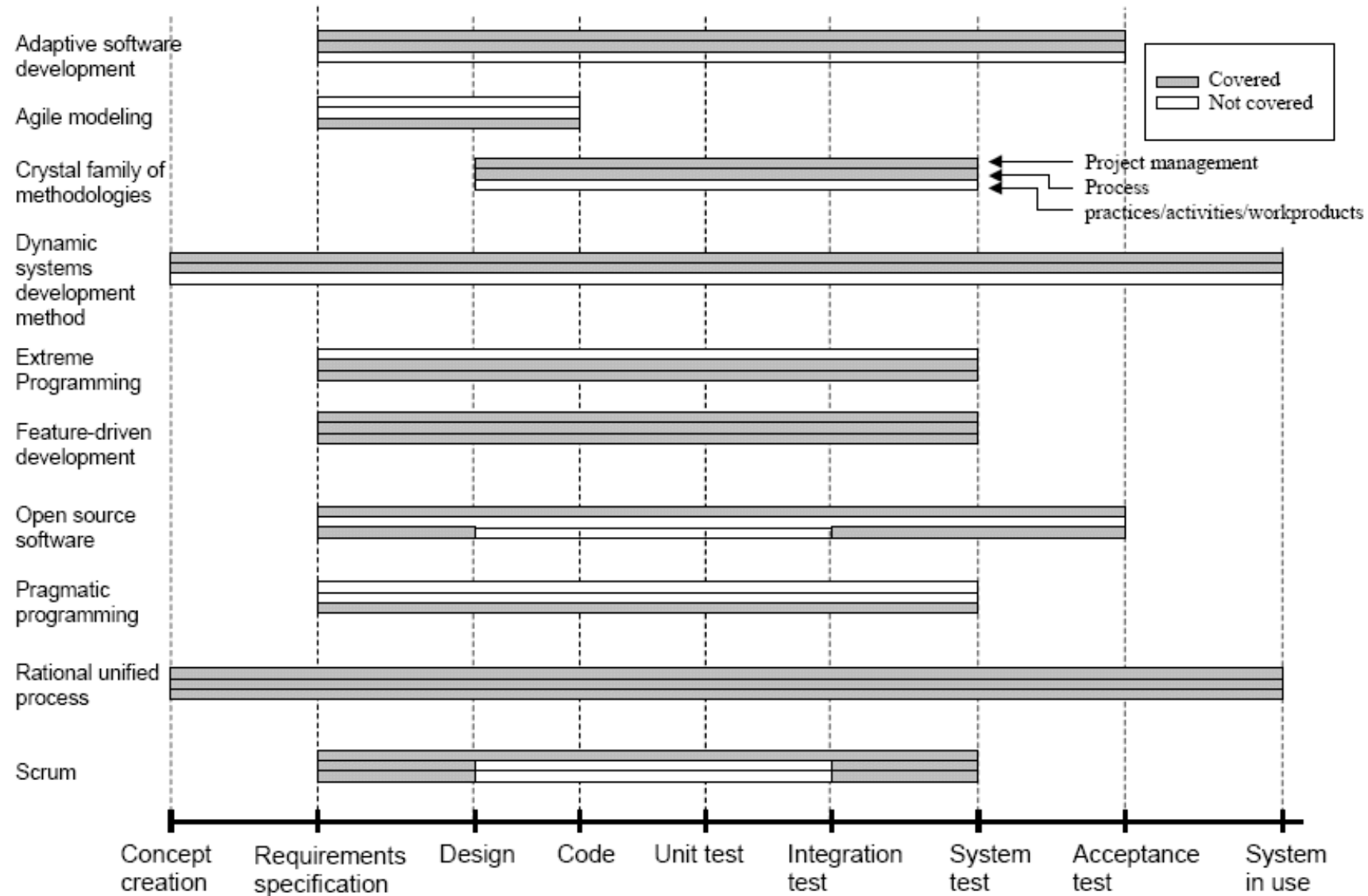
***Responding to change*** over following a plan

*That is, while there is value in the items on the right,  
we value the items on the left more"*

Source: [www.agilemanifesto.org](http://www.agilemanifesto.org)

# Agile Methodologies (AM)

## Agile and SLC Phases Coverage



**Source:** Abrahamsson P., Salo O., Ronkainen J. & Warsta J., *Agile Software Development Methods. Review and Analysis*, VTT Publication # 478, 2002

# Agile Methodologies (AM)

## Agile and SPI Methods

ML	Sw-CMM KPA	Rating Compliance	
		XP	SCRUM
2	<b>RM</b> - Requirements management	😊	😊
2	<b>SPP</b> - Software project planning	😊	😊
2	<b>SPTO</b> - Software project tracking and oversight	😊	😊
2	<b>SSM</b> - Software subcontract management	😞	😞
2	<b>SQA</b> - Software quality assurance	😐	😊
2	<b>SCM</b> - Software configuration management	😐	😐
3	<b>OPF</b> - Organization process focus	😐	😐
3	<b>OPD</b> - Organization process definition	😐	😐
3	<b>TP</b> - Training Program	😞	😞
3	<b>ISM</b> - Integrated software management	😞	😞
3	<b>SPE</b> - Software product engineering	😊	😊
3	<b>IC</b> - Intergroup co-ordination	😊	😊
3	<b>PR</b> - Peer reviews	😊	😞
4	<b>QPM</b> – Quantitative Project Management	😞	😞
4	<b>SQM</b> – Software Quality Management	😞	😞
5	<b>DP</b> – Defect Prevention	😐	😞
5	<b>TCM</b> – Technology Change Management	😞	😞
5	<b>PCM</b> – Process Change Management	😞	😞
<b>Legend:</b>			
		😐	Partially addressed;
		😊	Largely addressed;
		😞	Not addressed

# Agenda



- **Introduction**
- **Agile Methodologies (AM)**
  - ✓ A Musical Analogy
  - ✓ The Agile Manifesto
  - ✓ Agile and SLC phase coverage
  - ✓ Agile and SPI Methods
- **Estimation Issues in an Agile Sw Development Project**
  - ✓ XP – Extreme Programming
  - ✓ SCRUM
  - ✓ Crystal Family
  - ✓ FDD – Feature Driven Development
  - ✓ DSDM – Dynamic Systems Development Method
  - ✓ Other studies - proposals
- **Discussion on Estimation practices in AM**
  - ✓ NFR – Non-Functional Requirements
  - ✓ Sizing Units
  - ✓ Historical Data
  - ✓ Standards
- **Conclusions**

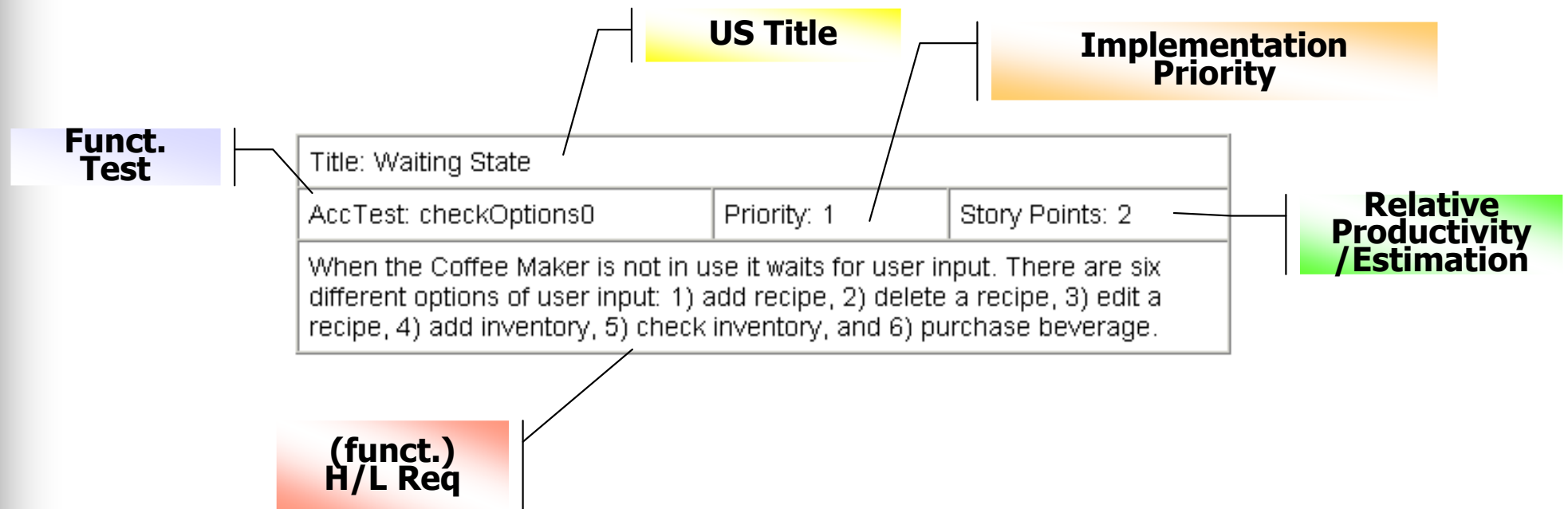
# Estimation Issues in AM

## XP – eXtreme Programming



- **(Functional) Requirement = User Stories (US)**

- ✓ Non-technical Language
- ✓ Focus only on the Functional Side of Requirements
- ✓ US is not a Use Case (UC)



Source: [open.ncsu.edu/se/tutorials/coffee\\_maker/](http://open.ncsu.edu/se/tutorials/coffee_maker/)

# Estimation Issues in AM

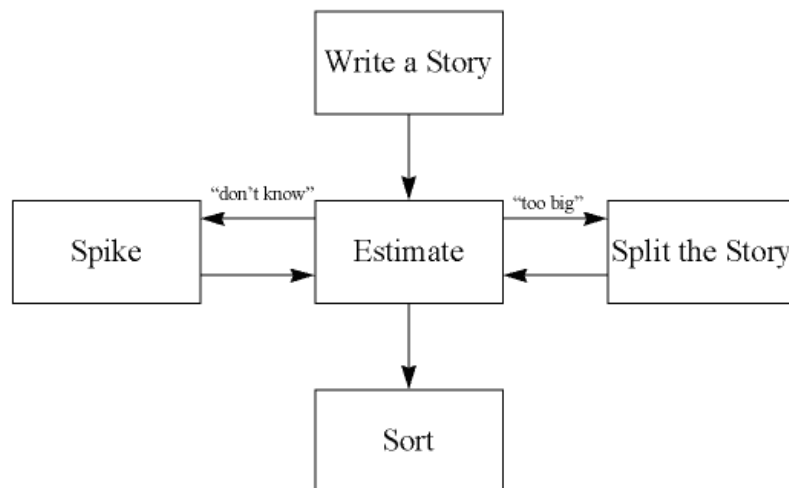
## XP – eXtreme Programming



- **Relative Productivity / Estimation concepts:**

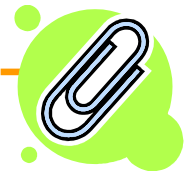
- ✓ **Story Point** → a (variable) no. of days needed to realize the US
- ✓ **Ideal Time** → "the time without interruption where you can concentrate on your work and feel fully productive"
- ✓ **Velocity** → "to measure the project velocity you simply count up how many user stories or how many programming tasks were finished during the iteration. Total up the estimates that these stories or tasks receive"
- ✓ **Load Factor** → "The load factor equals actual calendar days to complete a task divided by the developer's estimated "ideal" days to do it"

- **Planning Game**



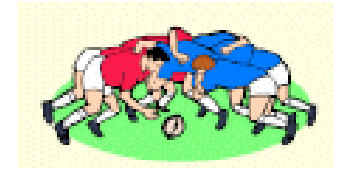
### Summarizing...:

- XP estimates by **experience/analogy**, estimating the effort needed mainly based on the request of series of functionalities

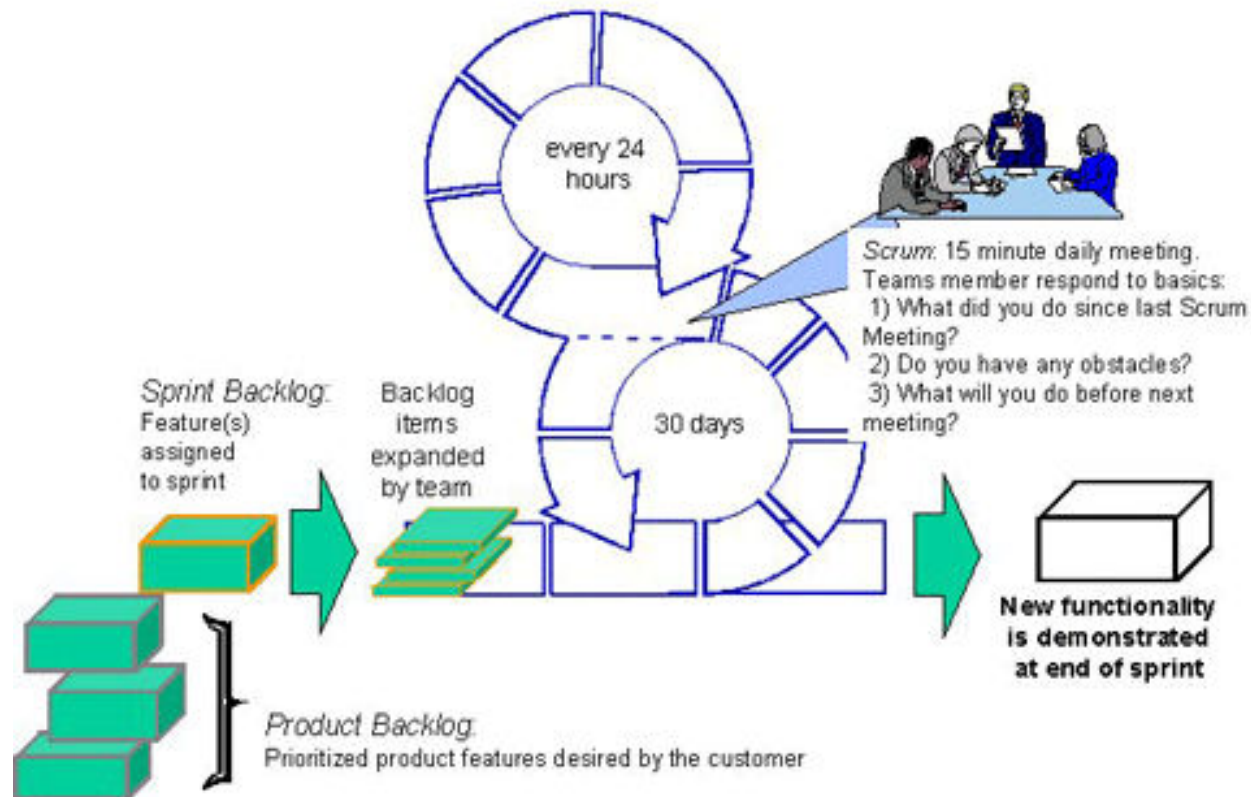


# Estimation Issues in AM

## SCRUM



- **From ASD to APM**
  - ✓ SCRUM is an Agile Project Management (APM) method
  - ✓ SCRUM is a general-purpose APM, not only for software projects!
- **The Process**



# Estimation Issues in AM

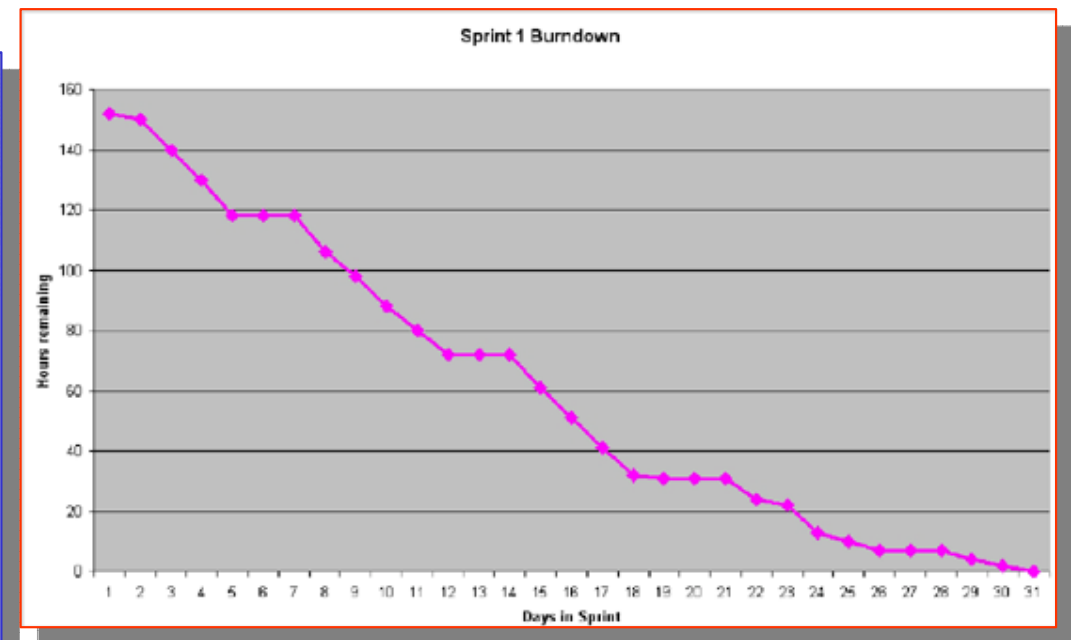
## SCRUM



- **Planning / Monitoring / Estimation concepts:**

- ✓ **Backlog** → two levels: **product** (*project*) & **Sprint** (*30-days iteration*)
- ✓ **Features** → User Req's to be allocated and worked
- ✓ **Micro-Planning** → within each Sprint, a WBS with detailed task
- ✓ **Daily Meeting** → the (daily) progress meeting for the project (risk mgmt)
- ✓ **Burndown chart** → a graph "burning" the total hours of a Sprint
- ✓ **NB:** no sizing measure!

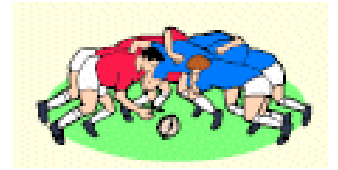
Product Backlog Estimating System Upgrade						
Sprint	ID	Backlog Item	Owner	Estimate (days)	Remaining (days)	
1	1	Minor	Remove user kludge in .dpr file	BC	1	1
1	2	Minor	Remove cMap/cMenu/cMenuSize from disciplines.pas	BC	1	1
1	3	Minor	Create "Legacy" discipline node with old civils and E&I content	BC	1	1
1	4	Major	Augment each tbl operation to support network operation	BC	10	10
1	5	Major	Extend Engineering Design estimate items to include summaries	BC	2	2
1	6	Super	Supervision/Guidance	CAM	4	4
7	Minor	Remove Custodian property from AppConfig class in globals.pas	BC	1		
8	Minor	Remove LOC_ constants in globals.pas and main.pas	BC	1		
9	Minor	New E&I section doesn't have lblCaption set	BC	1		
10	Minor	Delay in main.releaseform doesn't appear to be required	BC	1		
11	Minor	Undo modifications to Other Major Equipment in formExcel.pas	BC	1		
12	Minor	AJACS form to be centred on the screen	BC	1		
13	Major	Extend DUnit tests to all 40 disciplines	BC	6		



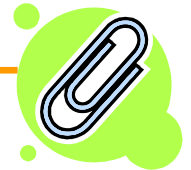


# Estimation Issues in AM

## SCRUM



### Summarizing...:



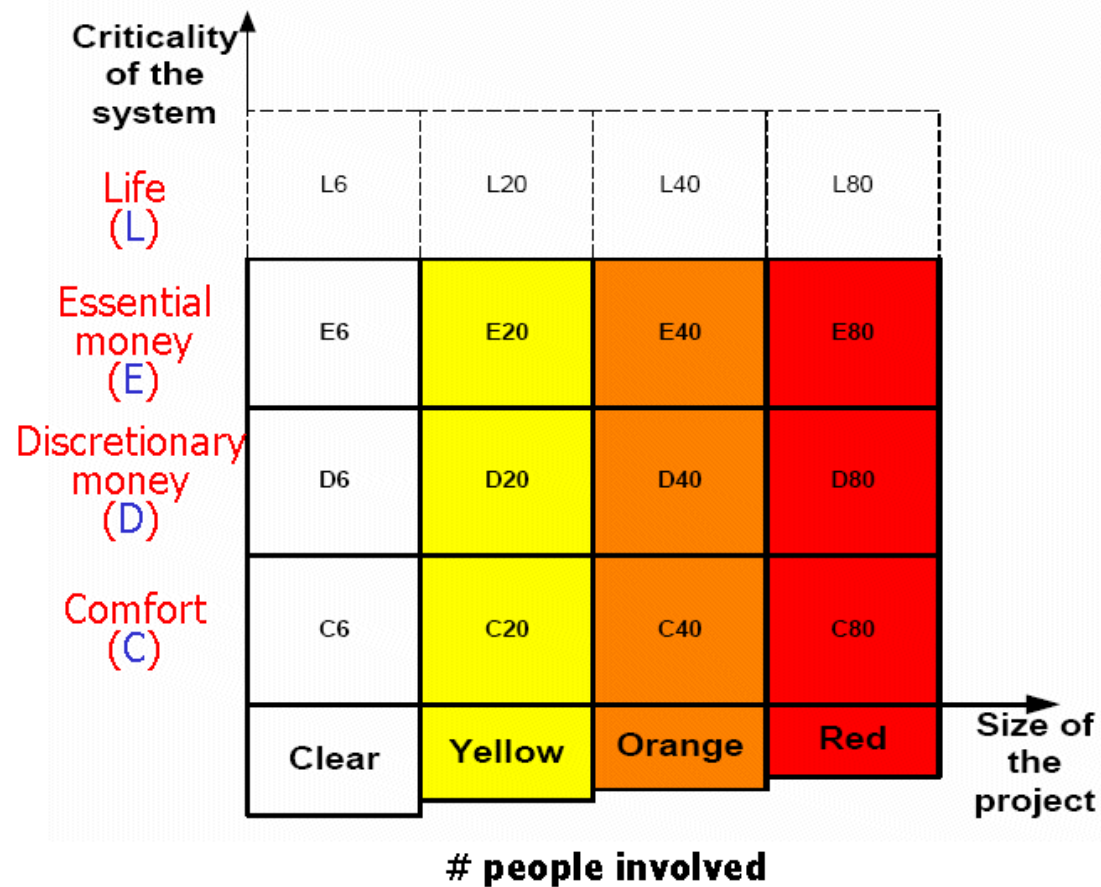
- SCRUM estimates by [experience/analogy](#), estimating directly the effort (in man-days or man-hours) for tracking both product and Sprints using the *burndown chart* that is, a graph showing how much time remains to complete the feature/tasks planned in such iteration, describing the project “velocity” the team is running.

# Estimation Issues in AM

## Crystal Family



- Crystal is a family of ASD
  - ✓ Two criteria: Size of the Project & Criticality of the System



# Estimation Issues in AM

## Crystal Family

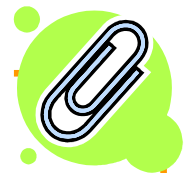


- **Planning / Monitoring / Estimation concepts:**

- ✓ **Blitz Planning (technique #3)** → the Crystal equivalent of XP User Stories, called **Blitz Planning Cards**
- ✓ **Delphi Analysis (technique #4)** → asd
- ✓ **Burn Charts (technique #9)** → both *burn-up* (as in EVMS) and *burn-down* (as in Scrum)
- ✓ **Ideal Days** → estimated as in XP

### Summarizing...:

- Crystal estimates by **experience/analogy**, estimating directly the effort (in ideal days to be refined into Estimated Elapsed Days) using Delphi analysis by some experts in the domain on the basis of counting content elements of such specific task (i.e. number of screens, business classes, ...).
- The project progress is tracked, as in Scrum, with “Burn” Charts



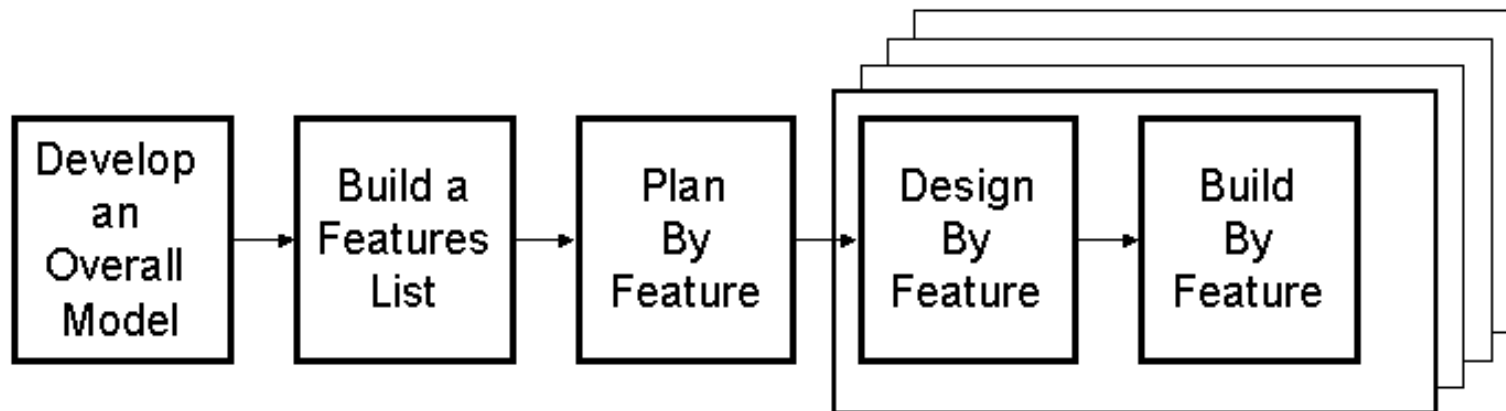
# Estimation Issues in AM

## FDD – Feature Driven Development

- **From ASD to APM**

- ✓ Structured into a series of processes (current version: **v1.3**)
- ✓ **Process #2**: Build the (functional-business) features list
- ✓ **Process #3**: Plan by Feature

- **The Overall Process**



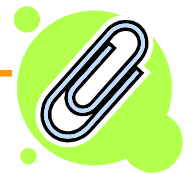
# Estimation Issues in AM

## FDD – Feature Driven Development

- **Planning / Monitoring / Estimation concepts:**
  - ✓ **Feature** → User Requirement
  - ✓ **Feature Duration** →
    - ✓ no longer than 2 weeks each to be completed
    - ✓ If longer, to be split into more granular ones
  - ✓ **Plan by Feature** → Process #3: it includes completion dates for each planned feature (month/year), properly assigned to team members

### Summarizing...:

- FDD estimates by **experience/analogy**, estimating directly the effort (in man-days or man-hours) for tracking planned features



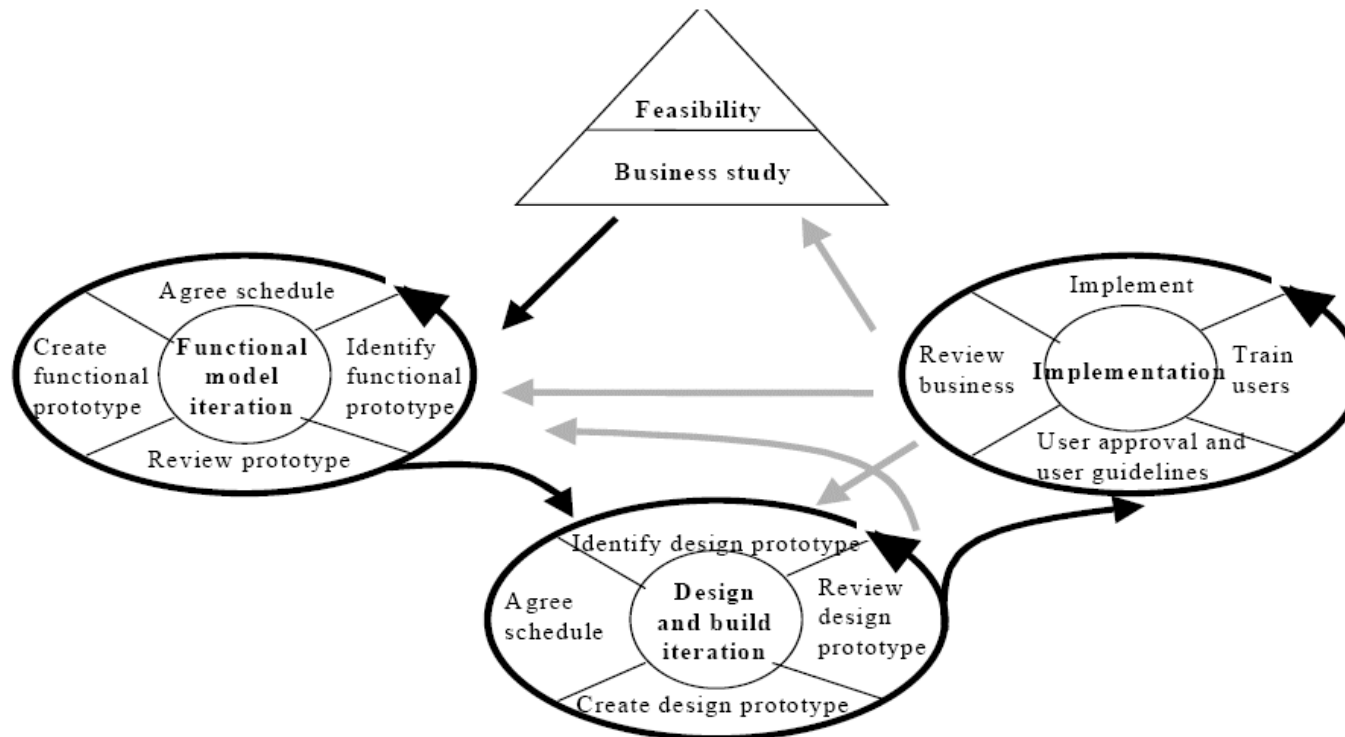
# Estimation Issues in AM

## DSDM – Dynamic Systems Development Method

- **SLC Characteristics**

- ✓ Three Main Stages (Pre-Project; Project; Post-Project)
- ✓ Several Iterations within the project lifecycle
- ✓ Prototype-based method

- **The Process**



# Estimation Issues in AM

## DSDM – Dynamic Systems Development Method



- **Planning / Monitoring / Estimation concepts:**

- ✓ **NFR** → DSDM takes care also of Non-Functional Req's
- ✓ Four major (estimation) phases (*Pre-Project, Planning at Project Start, (re)Planning during the Project lifetime, Planning at the increment end*)
- ✓ Usage of **Contingencies**
- ✓ Two estimation approaches:
  - ✓ **Top-Down**: based on business requirements, an early estimate is derived from high-level requirements, from a very high-level view on the project (i.e. estimation by analogy).
  - ✓ **Bottom-up**: based on tangible systems, figuring out for low-level components, take time to be prepared, need sufficiently detailed information, provide a good basis for planning the project (i.e. Function Point Analysis, Estimating from System Components, Collaborative Estimating).

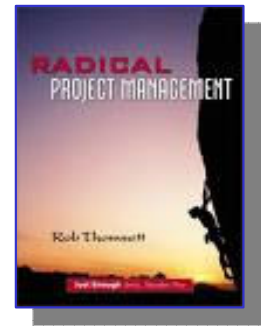
### Summarizing...:

- DSDM estimates by **experience/analogy**, estimating directly the effort (in man-days or man-hours) for tracking planned features **but** taking care of NFR **and** data gathering ("*it is essential to collect metrics from projects so that they can be validated and refined for use in estimating future projects, i.e. to provide continuous improvement of the estimating process*")



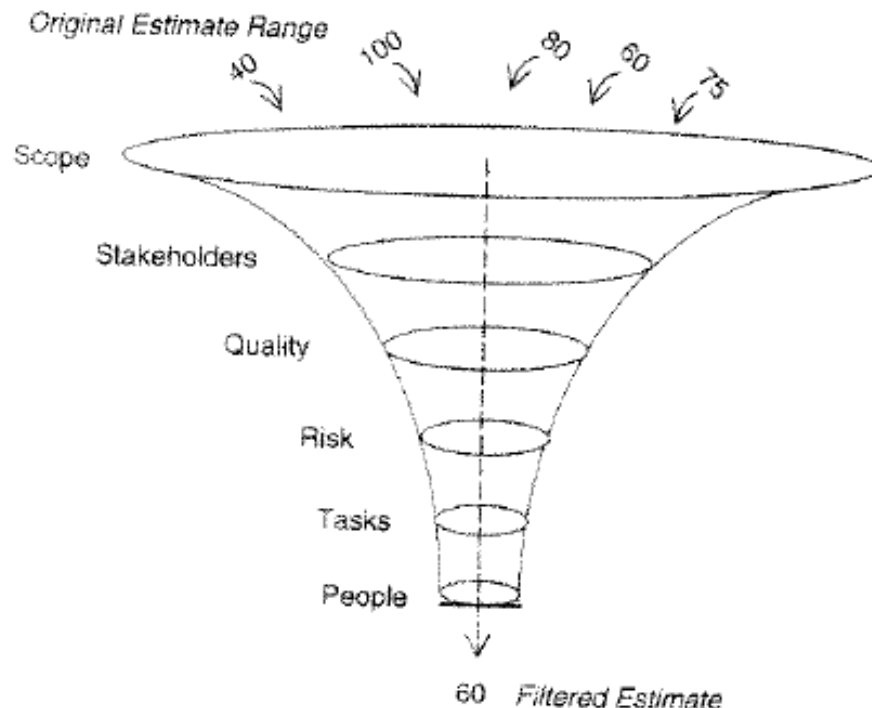
# Estimation Issues in AM

## Other Studies - Proposals



- **The Funnel of Increasing Accuracy (R.Thomsett)**

- ✓ "Radical" Project Management approach
- ✓ WBS-based, through wide-band Delphi estimates and Sensitivity Analysis (best, likely, worst cases)
- ✓ Estimation "adjusted" according to quality's requirements



### Summarizing...:

- Radical PM estimates always by **experience/analogy**, estimating directly the effort (in man-days or man-hours) **but** takes care of quality requirements as further filter towards the final estimated value



# Agenda



- **Introduction**
- **Agile Methodologies (AM)**
  - ✓ A Musical Analogy
  - ✓ The Agile Manifesto
  - ✓ Agile and SLC phase coverage
  - ✓ Agile and SPI Methods
- **Estimation Issues in an Agile Sw Development Project**
  - ✓ XP – Extreme Programming
  - ✓ SCRUM
  - ✓ Crystal Family
  - ✓ FDD – Feature Driven Development
  - ✓ DSDM – Dynamic Systems Development Method
  - ✓ Other studies - proposals
- **Discussion on Estimation practices in AM**
  - ✓ NFR – Non-Functional Requirements
  - ✓ Sizing Units
  - ✓ Historical Data
  - ✓ Standards
- **Conclusions**

# Discussion Estimation Practices in AM

## Main (missing) points identified



### ① NFR: Non-Functional Requirements

- ✓ the focus of AM on evaluating features, user stories, or more general high-level user requirements mainly from a functional viewpoint **does not include** the non-functional perspective of the projects.
- ✓ explicitly, only few methods such as **DSDM** propose in their practices to consider them for estimation and planning purposes.

### ② Sizing Units

- ✓ general AM approach to estimation experienced-based, **mainly** founded on functional analysis and **not** supported by historical data.
- ✓ **effort** and **schedule** are the time-related expressions corresponding to “how much” work will be performed; they do not mention the functional requirements measurement with a recognized quantitative measurement method. In this AM approach to estimation, **size** is only considered as “implicit” in the estimators’ minds.
- ✓ **Story Points** and **Velocity** concepts represent a “**delivery rate**” based only on the experience and knowledge of such particular team, applicable with success in small and stable teams. But it is much **more challenging in larger organizations** where the composition of a project team is not necessarily as stable over time.

# Discussion Estimation Practices in AM

## Main (missing) points emerged



### ③ Historical Data

- ✓ the practice to collect data from projects is not required by most AM (except for DSDM).

### ④ Standards

- ✓ each methodology and each team applying a certain AM uses its own definitions;
- ✓ adequate measurements require to be based on standards and require inputs with a validated, quality level.

# Agenda



- **Introduction**
- **Agile Methodologies (AM)**
  - ✓ A Musical Analogy
  - ✓ The Agile Manifesto
  - ✓ Agile and SLC phase coverage
  - ✓ Agile and SPI Methods
- **Estimation Issues in an Agile Sw Development Project**
  - ✓ XP – Extreme Programming
  - ✓ SCRUM
  - ✓ Crystal Family
  - ✓ FDD – Feature Driven Development
  - ✓ DSDM – Dynamic Systems Development Method
  - ✓ Other studies - proposals
- **Discussion on Estimation practices in AM**
  - ✓ NFR – Non-Functional Requirements
  - ✓ Sizing Units
  - ✓ Historical Data
  - ✓ Standards
- **Conclusions**

# Conclusions



- Agile Methods (AM) represent interesting solutions for projects with unstable requirements, iterative SLC, short-term milestones and small teams.
- Only in the last five years, attention was shifted from ASD to APM (*Agile Project Management*). Through this interest, planning and scheduling practices have been tailored to agile methodologies, but with much less attention to the estimation process.
- AM still being a young approach to software development, much work remains to be done to improve the way AM manage estimates, including tailoring relevant practices from well-established and proven “heavyweight” methodologies.
- Next steps will include analyses of the impact on size, effort and productivity of non-functional requirements in agile projects to improve estimates right from the Requirement Elicitation phase using, for example, a new User Story structure called US<sup>2</sup> (2<sup>nd</sup> generation of User Stories).

*Coby's Observation:*

*Math is a fact, but problem data quite often are not “facts”*

# Q & A



# Grazie! Thank you!



[luigi.buglione@computer.org](mailto:luigi.buglione@computer.org), [alain.abran@etsmtl.ca](mailto:alain.abran@etsmtl.ca)