



**“MANAGEMENT IS
(FOR) THE DARK
SIDE.”**

**Unknown author
(with a black helmet)**

FEW WORDS ABOUT ME

- ▶ Educated as a theoretical particle physicist
- ▶ PhD thesis in CP3 (2003-2008)
- ▶ PostDoc at Nikhef (2008-2010)
- ▶ Worked in management consulting (McKinsey, 2010-2011)
- ▶ Co-founded consulting firm named B12 based in LLN (2012-...)



“MANAGEMENT IS A DISTINCT PROCESS CONSISTING OF PLANNING, ORGANISING, ACTUATING AND CONTROLLING, PERFORMED TO DETERMINE AND ACCOMPLISH THE OBJECTIVES BY THE USE OF PEOPLE AND RESOURCES”

George R.Terry

IN PRACTICE, MANAGEMENT MIGHT INCLUDE...

- ▶ **Project** management
- ▶ **Team/people** management
- ▶ **Financial** management
- ▶ **Strategy** and business management
- ▶ **Staffing and career** management
- ▶ ... and many other things!

THOSE LECTURES ARE DESIGNED TO BE:

- ▶ Everything I wish I know before...
- ▶ Interactive
- ▶ Practical/applicable ("toolkit")
- ▶ Non exhaustive/specialised/academic
- ▶ Fairly subjective
- ▶ Full of anecdotes & personal insights
- ▶ and fun! (I hope...)





MANAGEMENT ESSENTIALS
FOR PHYSICISTS

EPISODE 1: PROJECT MANAGEMENT

MICHEL HERQUET - UCLOUVAIN - MARCH 2017

“PROJECT MANAGEMENT IS THE DISCIPLINE OF INITIATING, PLANNING, EXECUTING, CONTROLLING, AND CLOSING THE WORK OF A TEAM TO ACHIEVE SPECIFIC GOALS AND MEET SPECIFIC SUCCESS CRITERIA.”

Wikipedia

TODAYS AGENDA

- ▶ A brief history of project management
- ▶ The "Classical" approach
- ▶ The "Agile" way
- ▶ Specific methodologies & tools
- ▶ Tips & tricks

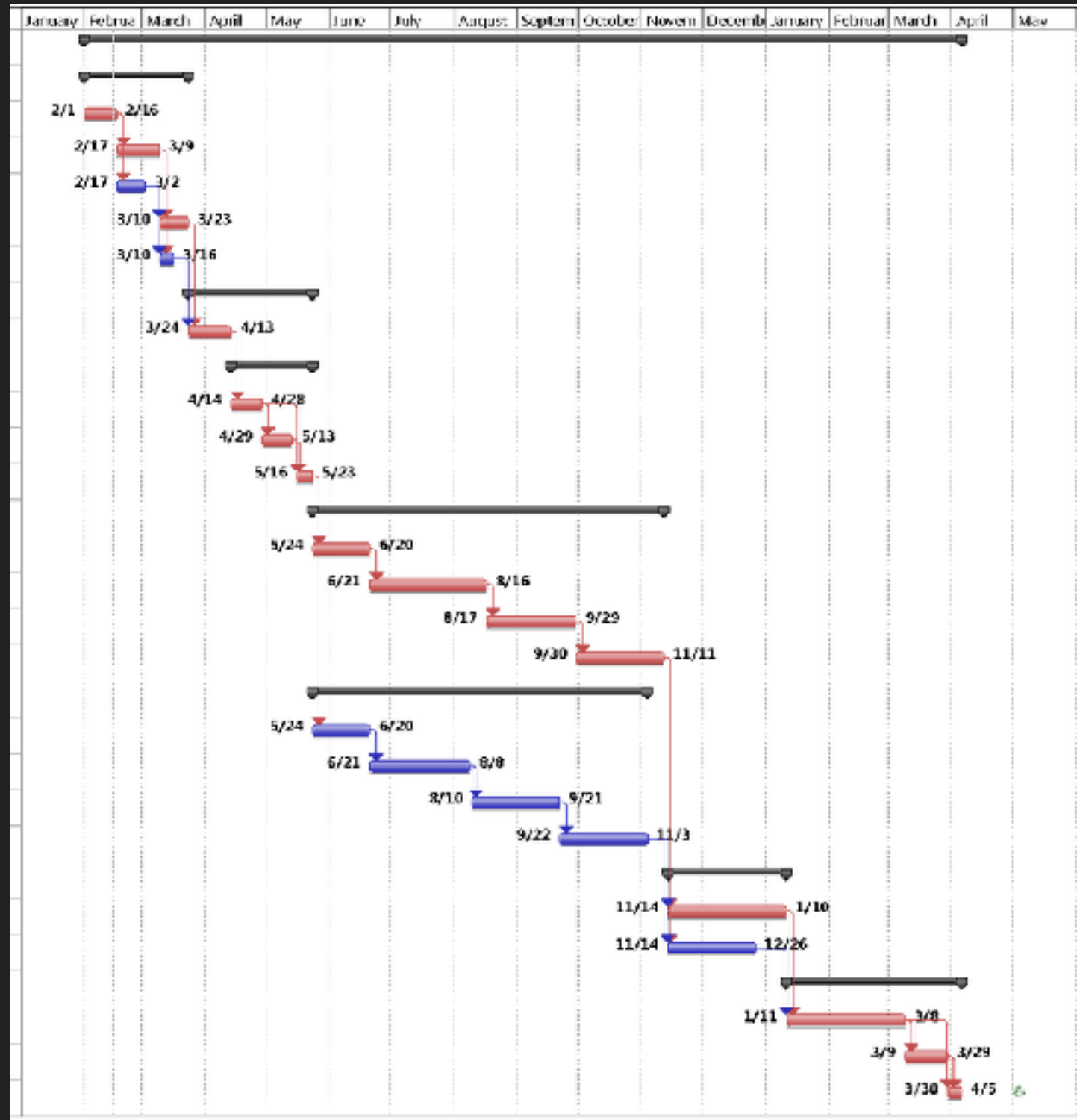


A FEW MILESTONES

- ▶ 2570 BC: The Great Pyramid of Giza is completed
- ▶ 1917: The Gantt chart is developed by Henry Gantt and applied to the Hoover Dam project
- ▶ 1957: The Critical Path Method (CPM) is invented by the Dupont Corporation and used for missile construction projects
- ▶ 1990: Scrum proposed as a project management style
- ▶ 1996: PRINCE2 Published by CCTA
- ▶ 2001: The Agile Manifesto
- ▶ 2012: ISO 21500 Standard for Project Management Released



THE CLASSICAL APPROACH



PLEASE DRAW ME A GANTT CHART

- ▶ A **Gantt chart** = showing activities (tasks or events) displayed against time
- ▶ Requires a **project plan**, a series of interdependent tasks that need to be performed in a particular order
- ▶ To create a project plan, split tasks into sub-tasks and creating a **task hierarchy**
- ▶ Adding dependency links (4 types), fixed constraints, fixed algorithms can determine **critical path(s)**
- ▶ Activities are classified as **critical** (fixed, part of critical path) or has **having float** (can be moved)

A CONCRETE EXAMPLE

- ▶ A project has been defined to contain the following list of activities along with their required times for completion.
 - ▶ Construct a Gantt chart for this project
 - ▶ Show the critical path
 - ▶ Determine the earliest expected completion time
 - ▶ What would happen if the screen design is revised to take 6 weeks instead of 2 weeks?

Activity No.	Activity	Time (weeks)	Immediate Predecessors
1	Collect requirements	3	
2	Analyze processes	2	1
3	Analyze data	2	2
4	Design processes	6	2
5	Design data model	3	3
6	Design screens	2	3,4
7	Design reports	4	4,5
8	Code	5	6,7
9	Test and Document	7	7
10	Deploy	2	8,9

AND HERE COME THE BAD NEWS...

- ▶ What if you cannot estimate the task lengths accurately upfront ?
- ▶ What if a task takes longer than expected and do affect the length of the critical path ?
- ▶ What if new tasks are added, and some removed ?
- ▶ What if there is simply no time for formal planning/reporting ?
- ▶ What if ... ?

**CLASSICAL MANAGEMENT
POORLY HANDLES UNCERTAINTY!**





BUILD A MANAGEMENT PHILOSOPHY BASED ON

- ▶ *Adaptability*: changes are **unavoidable**, goals and processes should be adapted continuously
- ▶ *Iterative progress*: processes are decomposed into fixed length **iterations**
- ▶ *Communication*: direct communication and good sense have **more value** than any other rule

A BRIEF HISTORY OF AGILITY

- ▶ Initially introduced in the context of software development, in opposition to “bridge building”
- ▶ First traces go back to 1960's (NASA Mercury program, IBM, etc.)
- ▶ Preliminary/partial approaches introduced in the early 1980's: RAD (rapid development), EVO (Evolutionary Delivery)



THE FOUNDING FATHERS

- ▶ 1990, *Scrum*, Ken Schwaber: scientific study of empirical management methods
- ▶ 1991, *Crystal*, Alistair Cockburn: interviews to determine common denominator of successful management styles
- ▶ 1996, *eXtreme Programming (XP)*, Kent Beck: "saving" a doomed payroll project



“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.”

The Agile Manifesto (2001)

SCRUM

- ▶ *Roles:*
 - ▶ **Product owner:** the “voice” of the end “user”
 - ▶ The product development **team**
 - ▶ The “**Scrum Master**”: a facilitator
- ▶ *Sprints:* iterations from one week to one month
- ▶ Formalised *key meetings* (daily scrum, sprint planning, sprint review, etc.)
- ▶ *Duality:* product backlog (user stories) and sprint backlog (tasks)





SCRUM PERSONAL ANALYSIS

▶ Advantages

- ▶ Very formalised (certified coaches, etc.)
- ▶ Well spread across functions and industries
- ▶ "Quantitative"

▶ Disadvantages

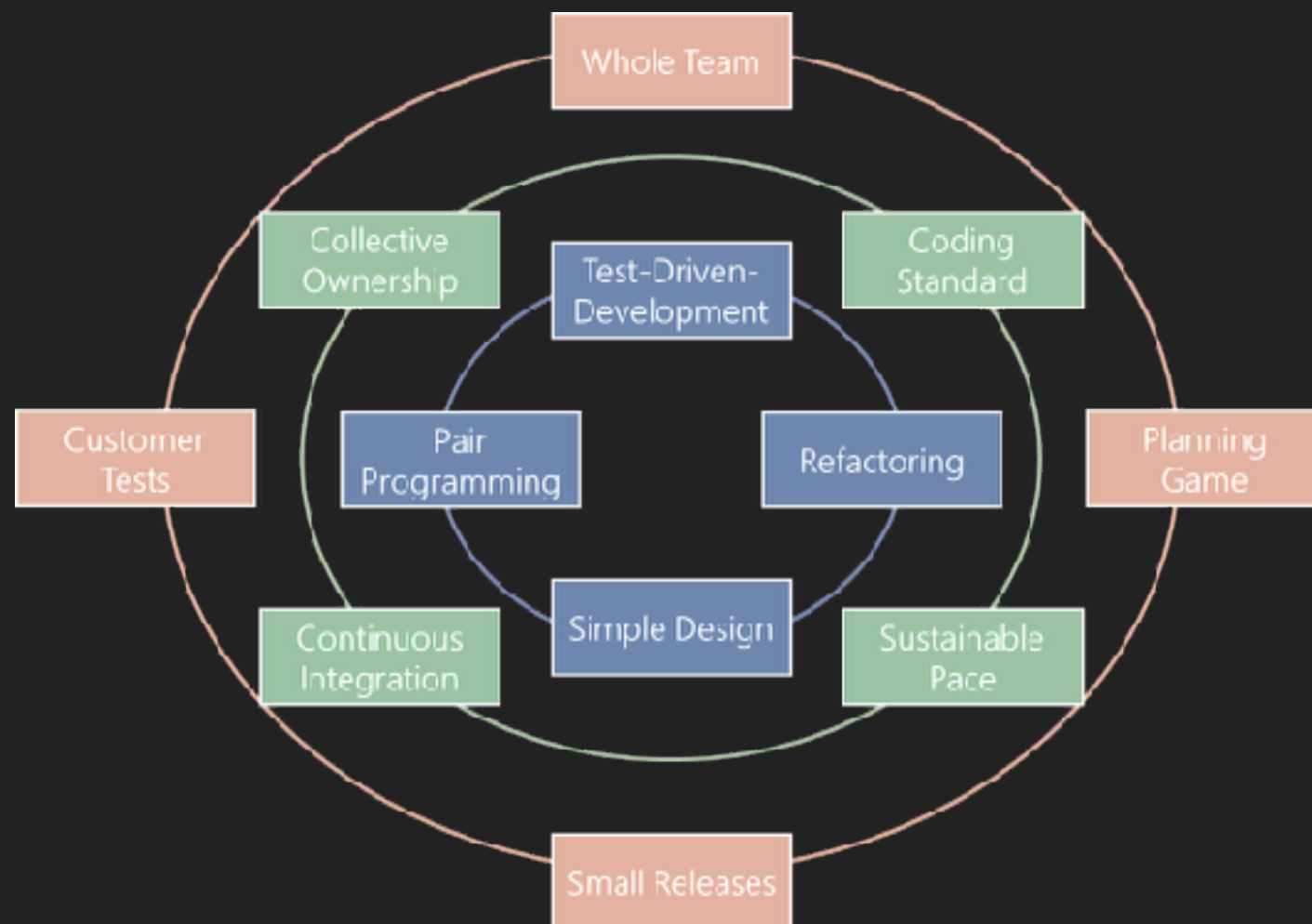
- ▶ Crucial "Scrum Master" role (single point of failure)
- ▶ Ressource demanding
- ▶ Very well known and often used to hide other issues

EXTREME PROGRAMMING (XP)

- ▶ Initially much focused on technical development
- ▶ **Philosophy** based on **values** (communication, simplicity, continuous improvement, courage, etc.)
- ▶ “Embracing changes”: change **does** create value
- ▶ List of interconnected “**principles**” (e.g., testing, peer working, best practices, continuous integration, refactoring, no extra hours, etc.)



XP PERSONAL ANALYSIS



► Advantages

- Strict principles, universal values
- Really open to change

► Disadvantages

- Crucial role for everybody!
- Not well suited for large structures (>50 people)
- Not as known as Scrum and much less spread

SCRUM/XP COMPARISON

Notion	Scrum	XP
Iteration length	Typically two weeks to a month	Typically a week, but a lot of interconnected time scales
Change during iteration	Not allowed	Allowed
Prioritisation	Based on product owner and team constraints	Only based on product owner constraints
Best Practices	Not really	Yes (a lot)

TIPS & TRICKS

- ▶ As for any management method, agile principles and best practices should be sustained over the **long term**
- ▶ The entire team needs to be **converted** and fully **convinced**
- ▶ Agile methodologies do have **drawbacks**, especially when confronted to classical approaches (budgeting, request for proposal, predefined milestones, etc.)
- ▶ The ultimate agility is to sometimes **go back to classical methods!**



PLAY IT AGILE!

- ▶ In group of 2-3: choose a common test project (paper to write, software to develop, report to publish, etc.)
 - ▶ Write down 10-15 "user stories"
 - ▶ Estimate their complexity (0, 1, 2 or 3)
 - ▶ Prioritise them
 - ▶ Group them into iterations
- ▶ Ask somebody from another group to challenge your plan
- ▶ Go to www.pivotaltracker.com, register for free and try the tool!

KEY TAKEAWAYS

- ▶ Project management is an established discipline with a long history and many methodologies
- ▶ Most of them can be classified as “classical” or “agile”, the latter being most efficient to handle evolving constraints
- ▶ What matters is to try and decide by yourself what works for you and your team, don't be afraid to **invent** your own method

**“YOU DON'T LEARN TO WALK BY
FOLLOWING RULES.
YOU LEARN BY DOING, AND BY FALLING
OVER.”**

Richard Branson