Software Development Methods and Some Other Experiences

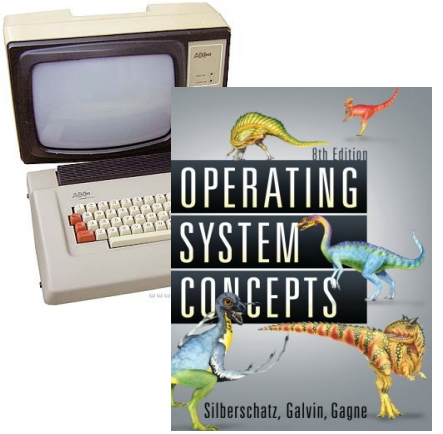
lars.karlsson@opnova.se

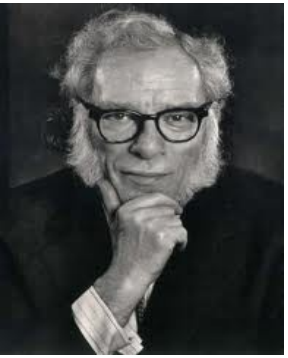
Agenda

- Lars Karlsson
- Methods
  - Waterfall
  - UP/RUP
  - ICONIX
  - SCRUM
- Some Personal Opinions & Experiences

Lars Karlsson

Lars Karlsson





Employment

South Tees Hospitals NHS
NHS Foundation Trust

1992-1993, Programmer

EC KONSULT

1993-2009, Software Engineer

Avalon INNOVATION
MANAGEMENT-TECHNOLOGY-FINANCE

2009-2012, Technical Manager, Software Engineer

OPNOVA

2012-, Self-employed, Software Engineer

Work Experience

First of all …The Customer

- Add value
  - By implementing a requirement specification ?
  - By working as a resource ?
  - By Helping, leading and/or directing ?

Some methods/processes

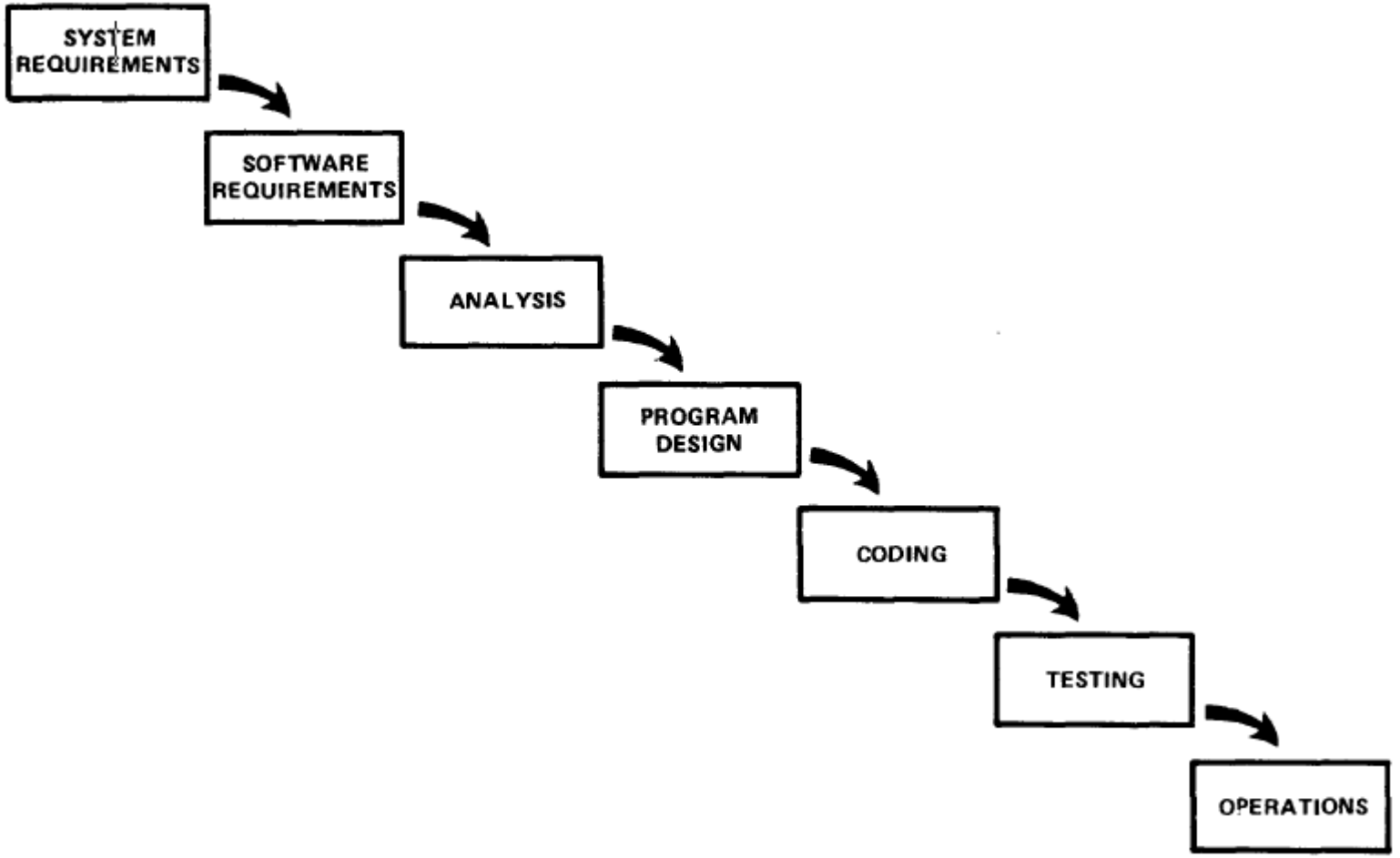
A methodology or a process...

isn't doing the thinking.

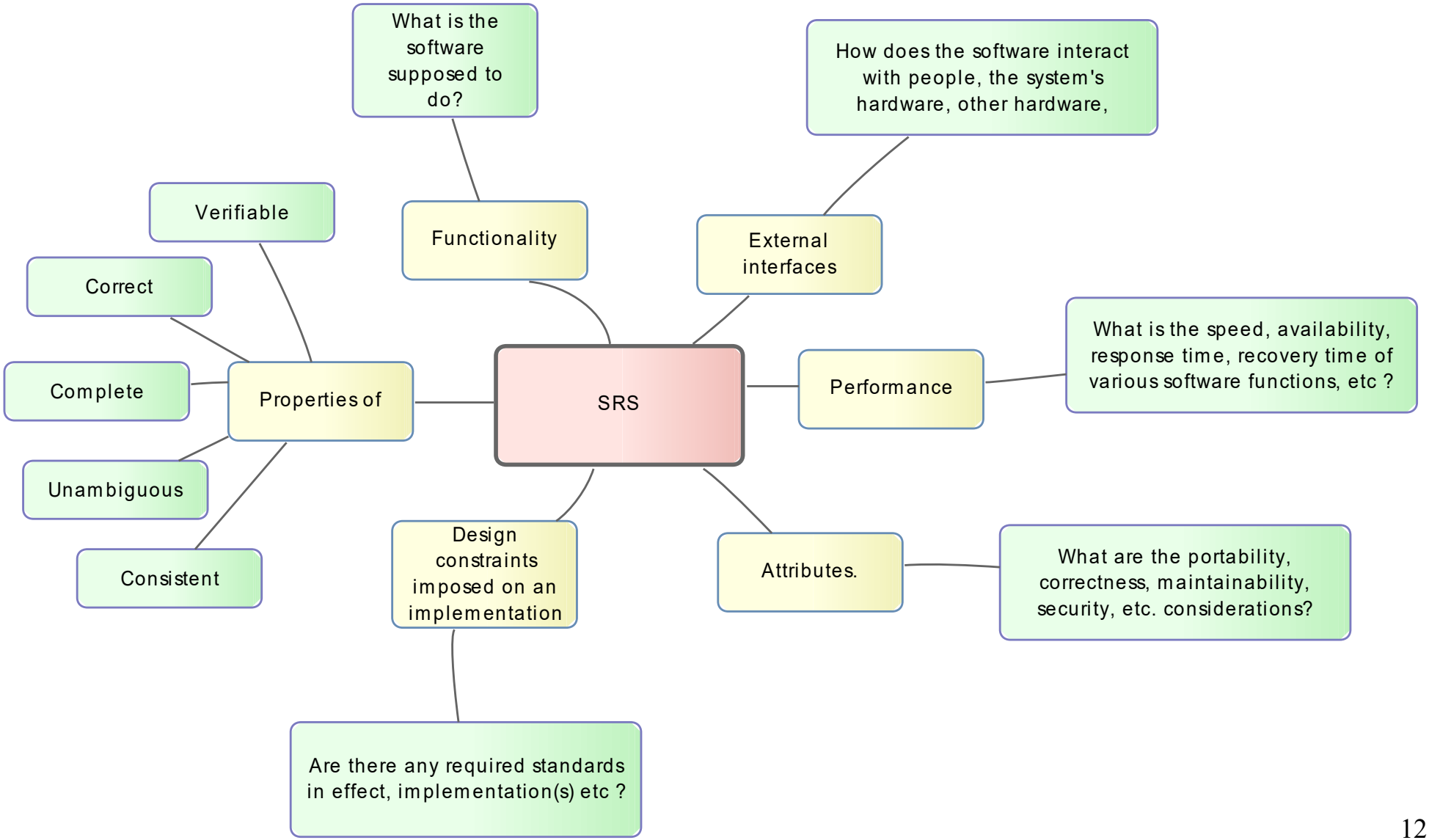
You are.

Waterfall Process

- Managing The Development Of Large Software Systems
  - Dr Winston W. Royce, 1970
- Sequential
- Many have gone bezerk on both the process and Dr Royce (including myself) , but ...
- .. there is a twist ...


SYSTEM REQUIREMENTS
SOFTWARE REQUIREMENTS
ANALYSIS
PROGRAM DESIGN
CODING
TESTING
OPERATIONS

Requirement Specification



SRS Content

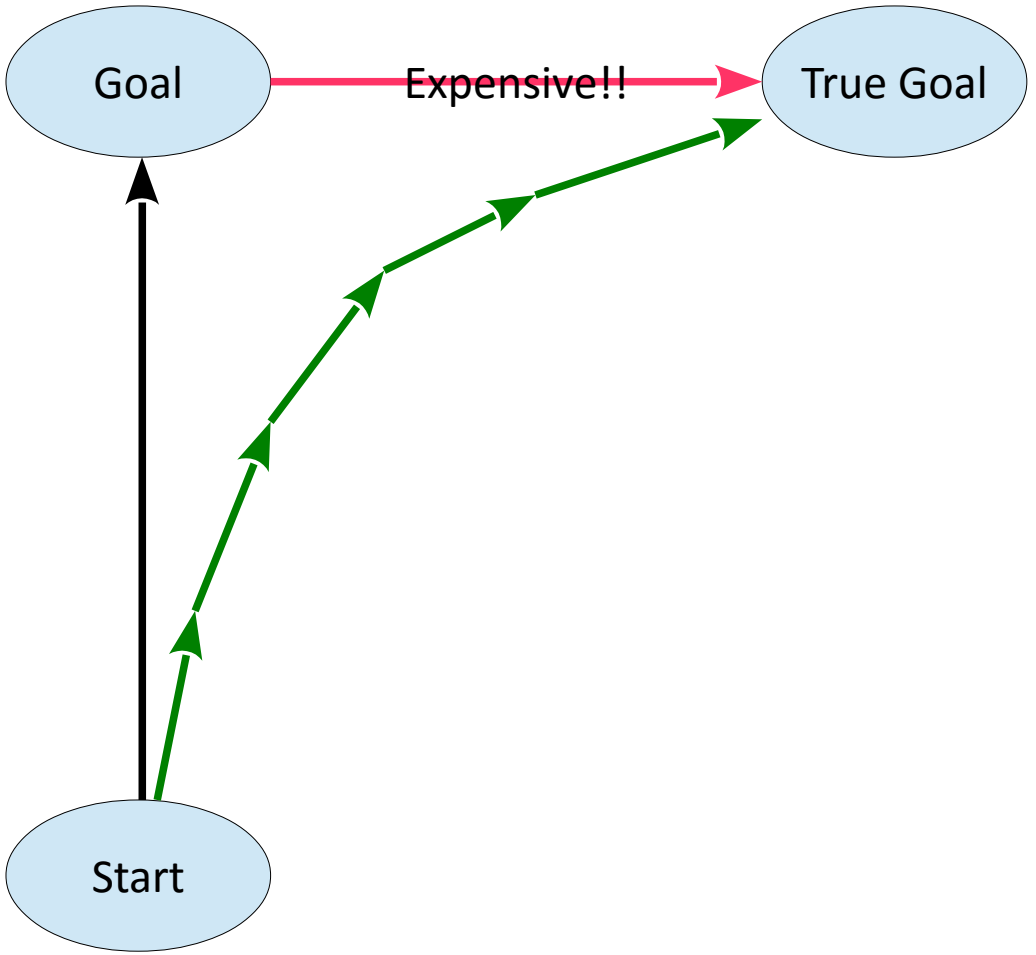
1. Introduction
1.1 Purpose
1.2 Scope
1.3 Definitions, Acronyms, and Abbreviations
1.4 References
1.5 Overview

2. General Description
2.1 Product Perspective
2.2 Product Functions
2.3 User Characteristics
2.4 General Constraints
2.5 Assumptions and Dependencies

4. Analysis Models
4.1 Sequence Diagrams
4.3 Data Flow Diagrams (DFD)
4.2 State-Transition Diagrams (STD)
5. Change Management Process

3. Specific Requirements
3.1 External Interface Requirements
3.1.1 User Interfaces
3.1.2 Hardware Interfaces
3.1.3 Software Interfaces
3.1.4 Communications Interfaces
3.2 Functional Requirements
3.2.1 <Functional Requirement or Feature #1>
3.2.2 <Functional Requirement or Feature #2>
3.3 Use Cases
3.3.1 Use Case #1
3.3.2 Use Case #2
3.4 Classes / Objects
3.4.1 <Class / Object #1>
3.4.2 <Class / Object #2>
3.5 Non-Functional Requirements
3.5.1 Performance
3.5.2 Reliability
3.5.3 Availability
3.5.4 Security
3.5.5 Maintainability
3.5.6 Portability
3.6 Inverse Requirements
3.7 Design Constraints
3.8 Logical Database Requirements
3.9 Other Requirements

On creating stuff



The twist

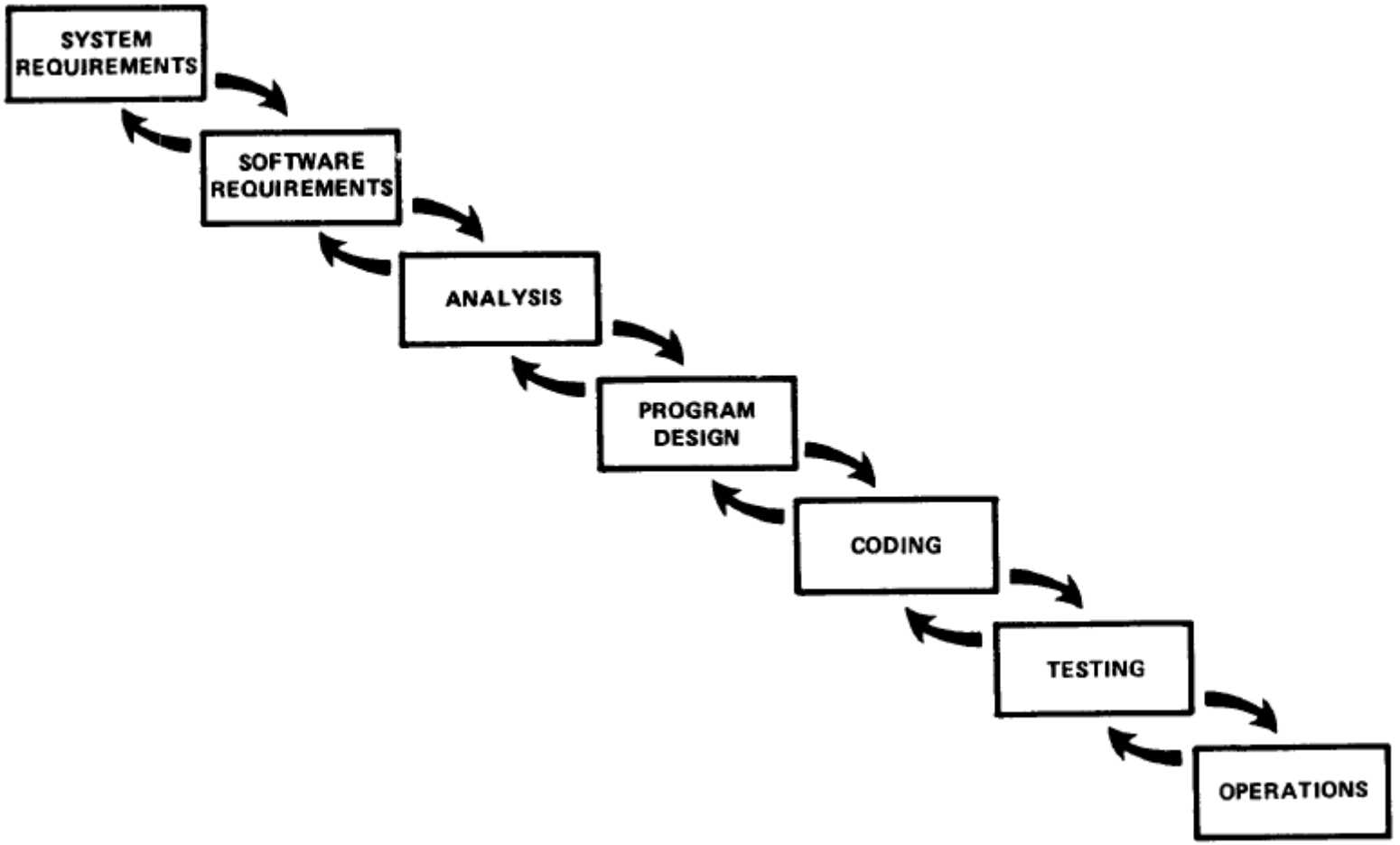
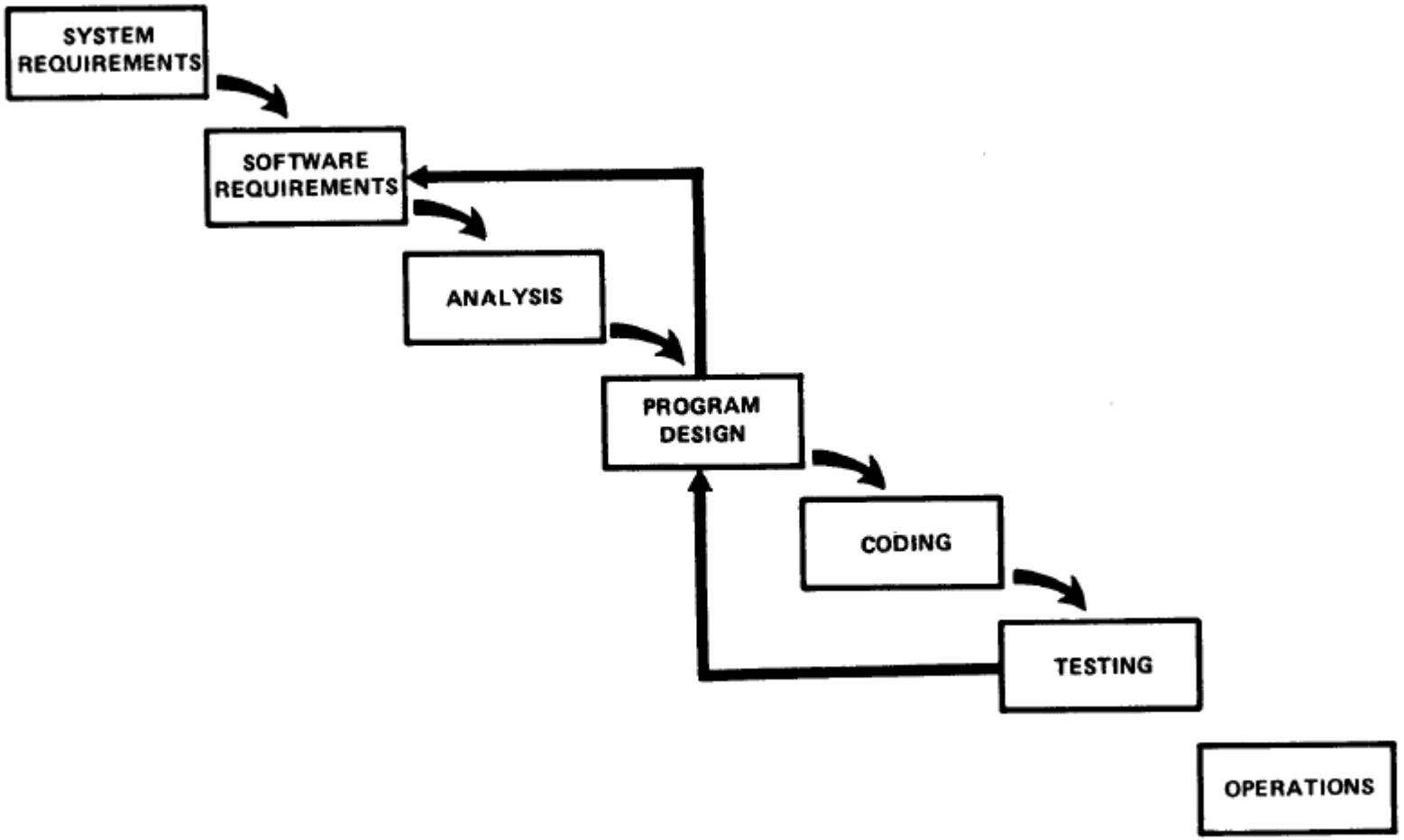


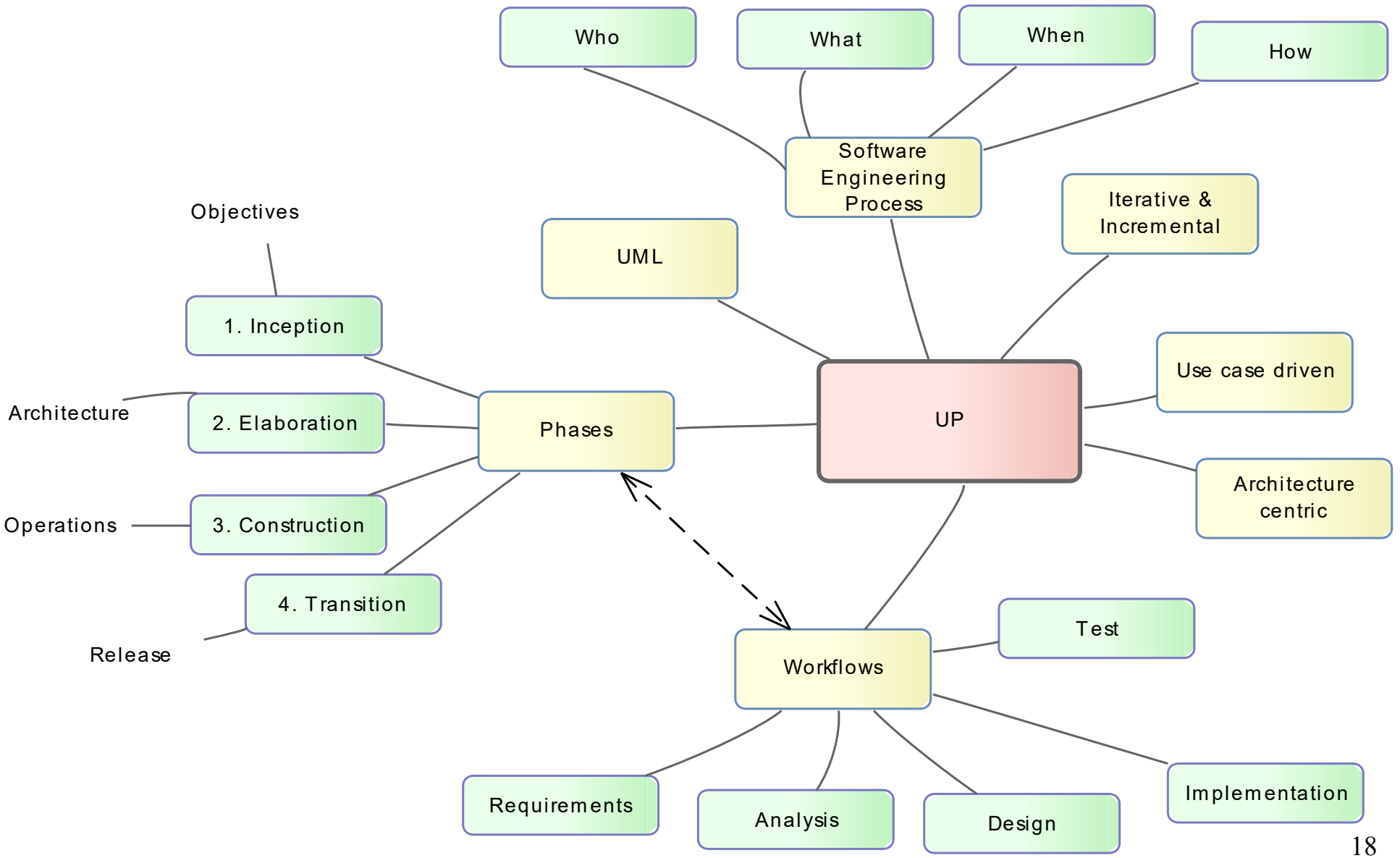
*Figure 3 portrays the **iterative relationship between successive development phases** for this scheme. The ordering of steps is based on the following concept: **that as each step progresses and the design is further detailed, there is an iteration with the preceding and succeeding steps** but rarely with the more remote steps in the sequence.*

*I believe in this concept, but the **implementation described above is risky and invites failure**. The problem is illustrated in Figure 4. The testing phase which occurs at the end of the development cycle is the first event for which timing, storage, input/output transfers, etc., **are experienced as distinguished from analyzed**.*

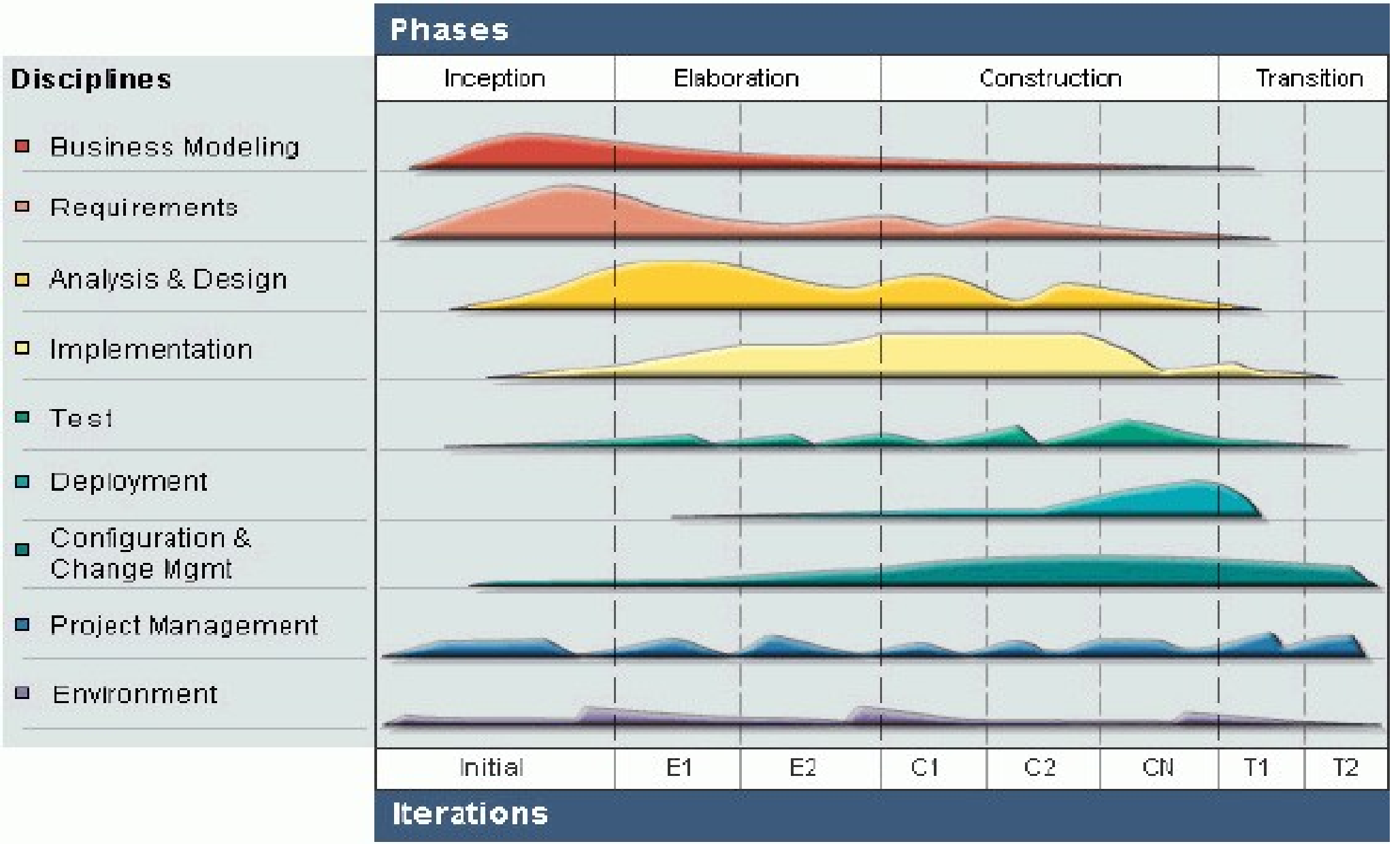


Unified Process

Unified Process



Unified Process



Manifesto for Agile Software Development

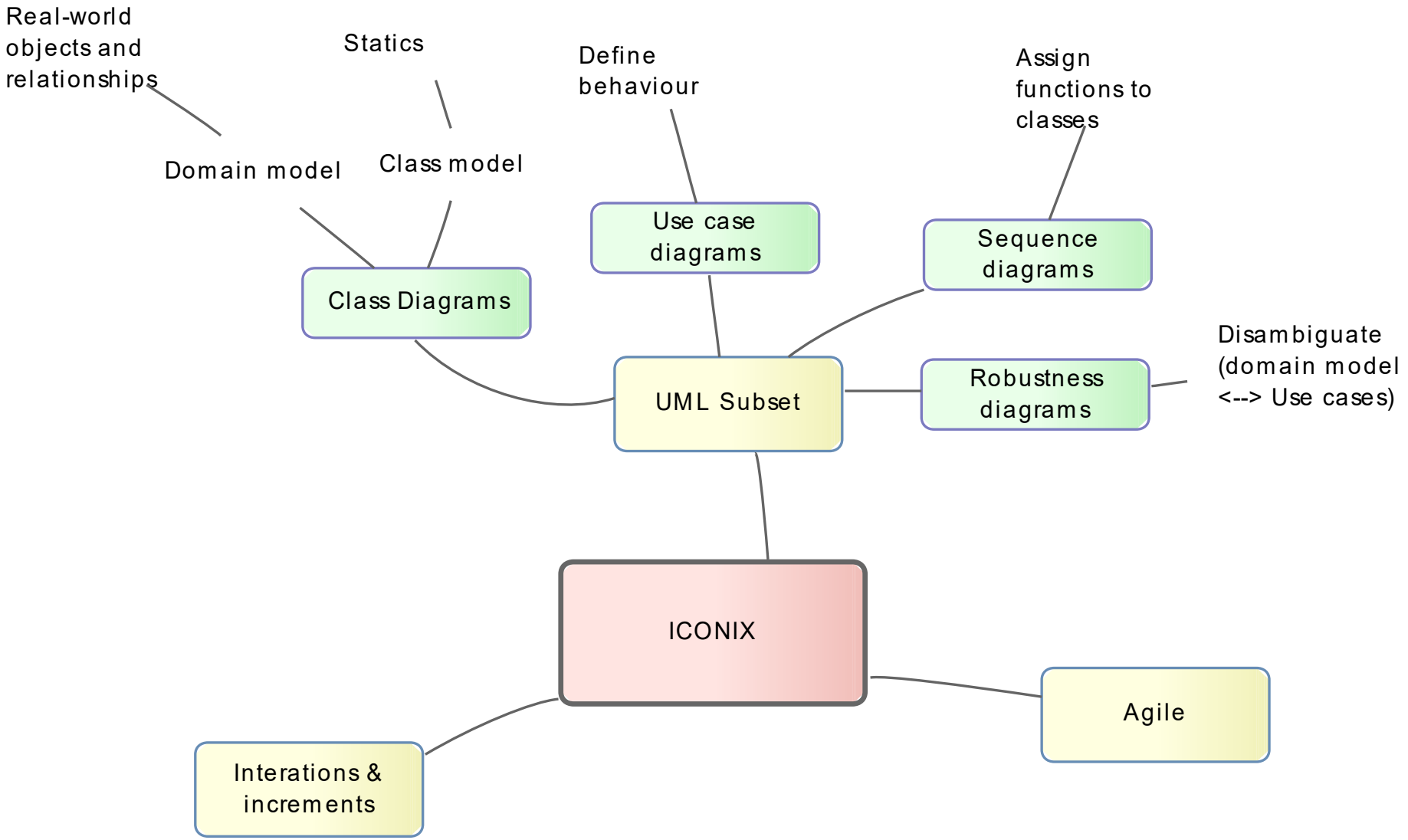
We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

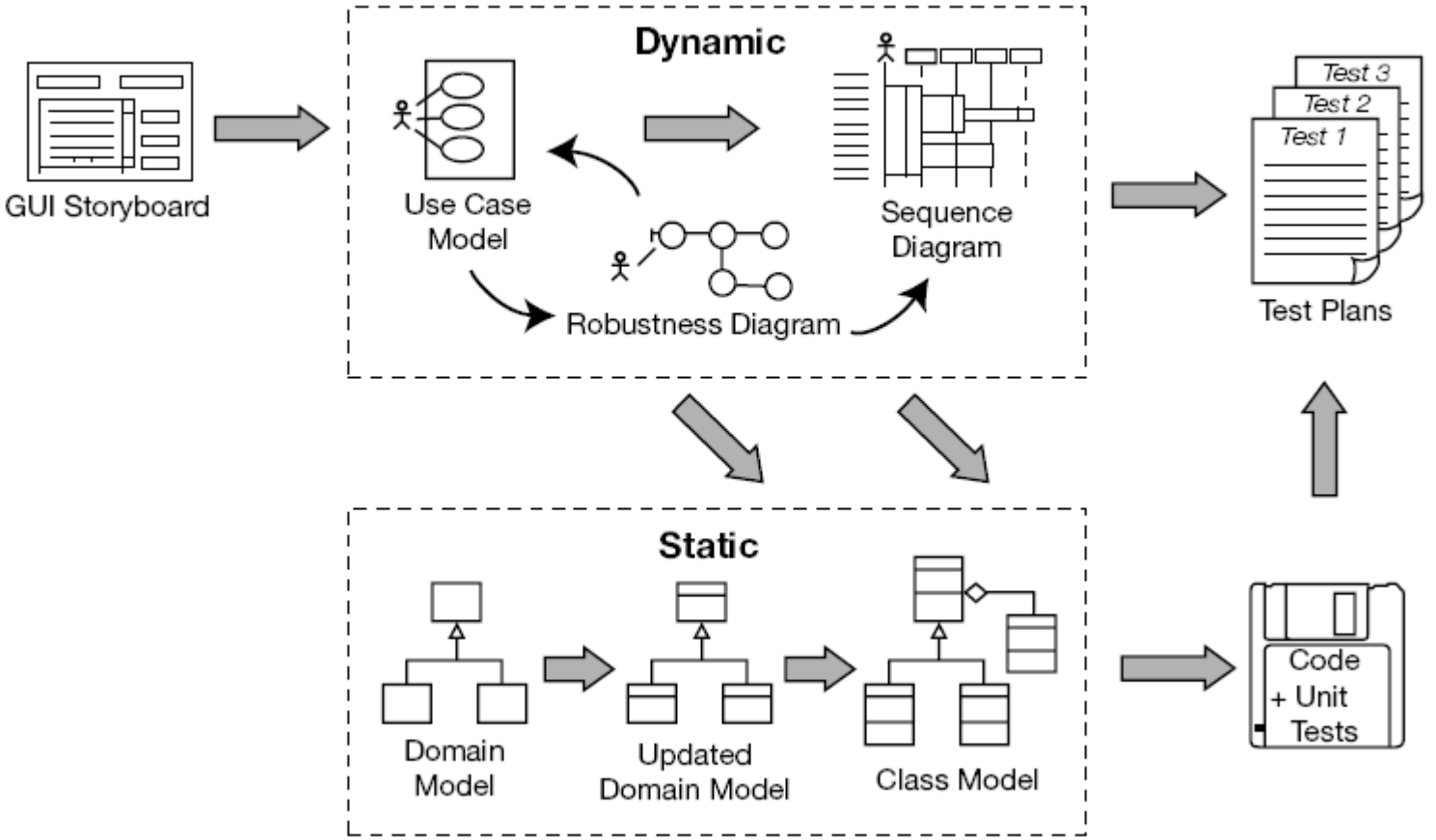
That is, while there is value in the items on
the right, we value the items on the left more.

ICONIX

ICONIX



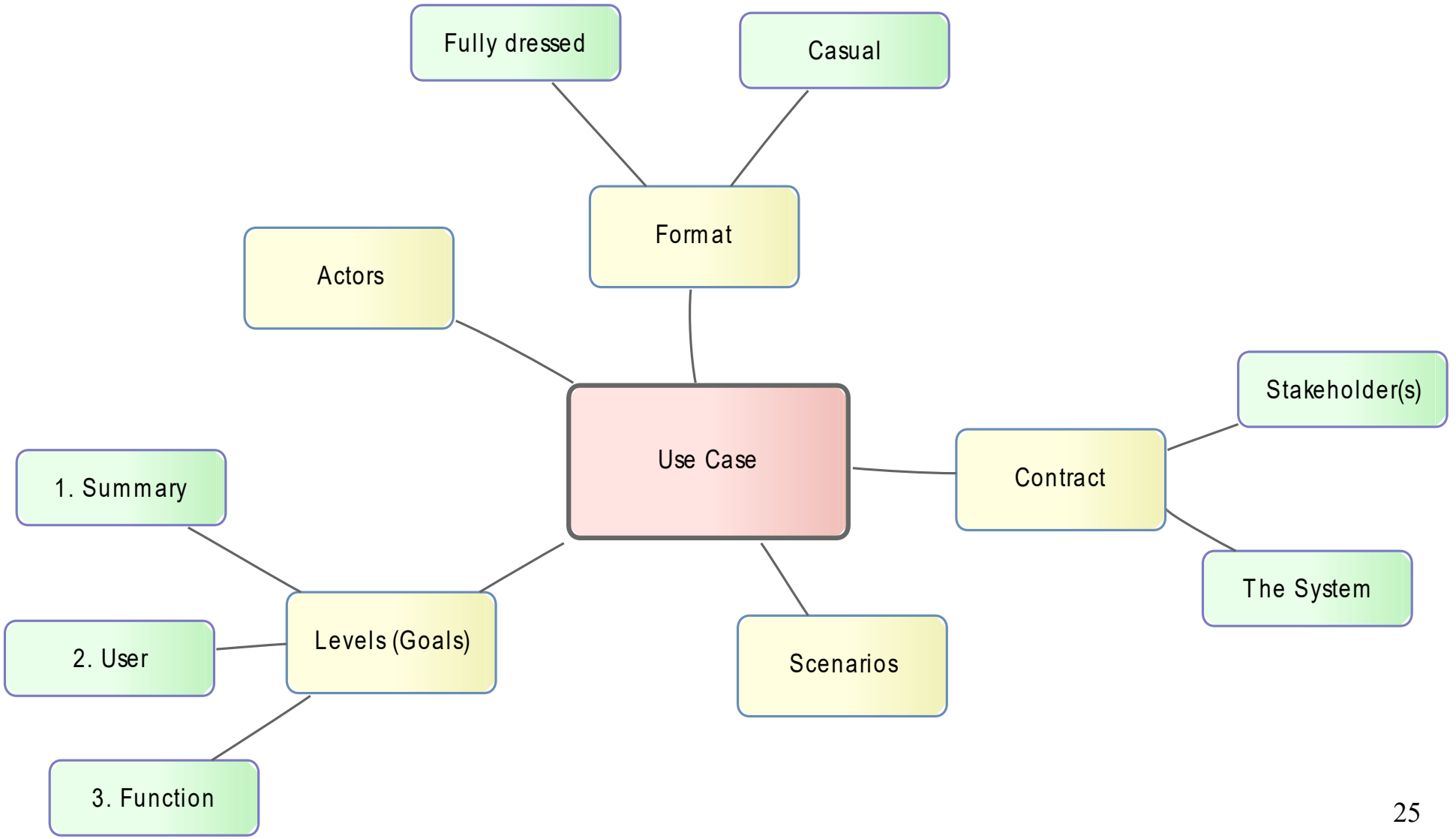
ICONIX



Domain model

- Real-world domain objects
- Glossary
- You and the customer
- Example : Monopoly

Use Case



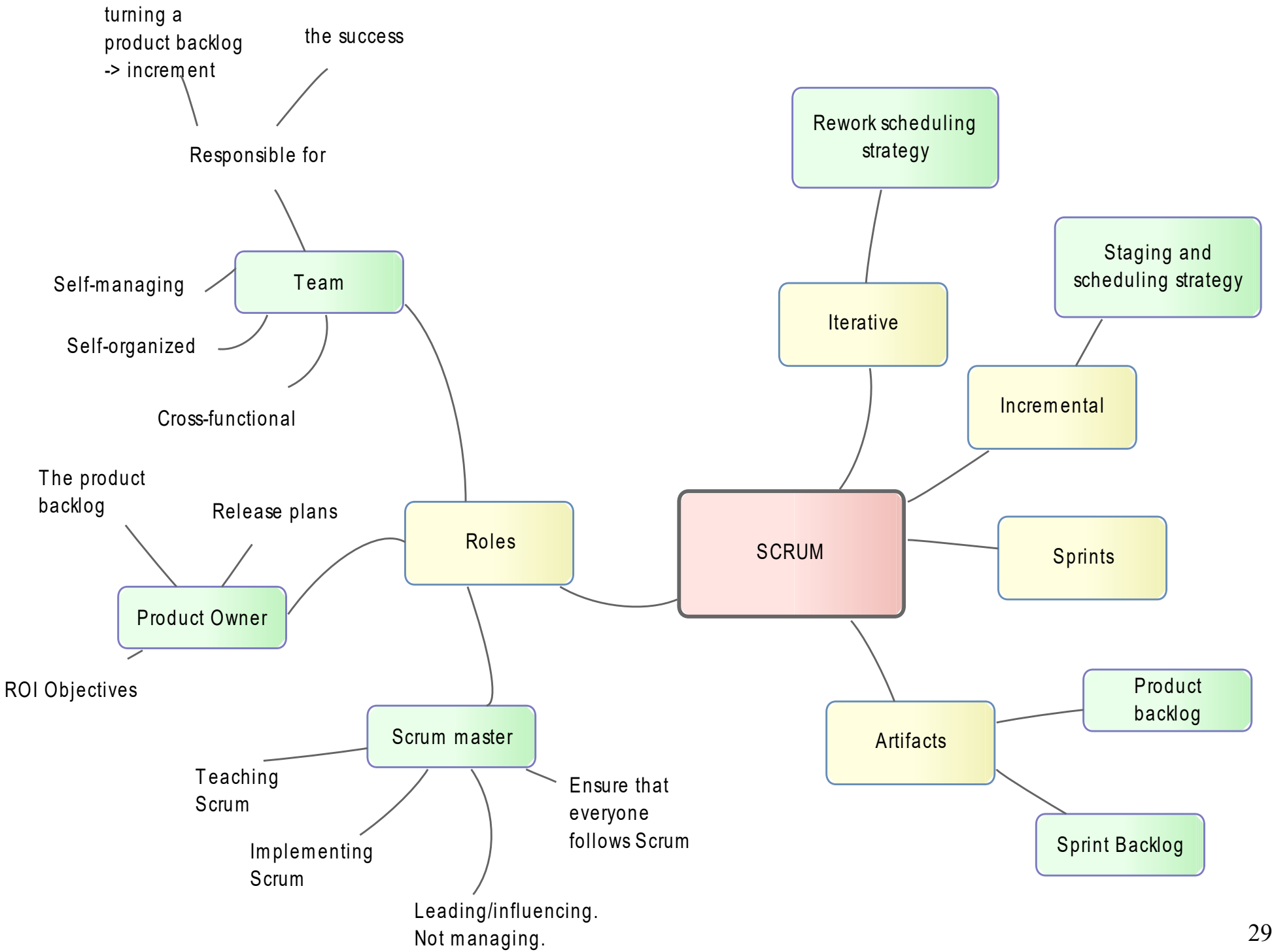
UC Template – Fully dressed

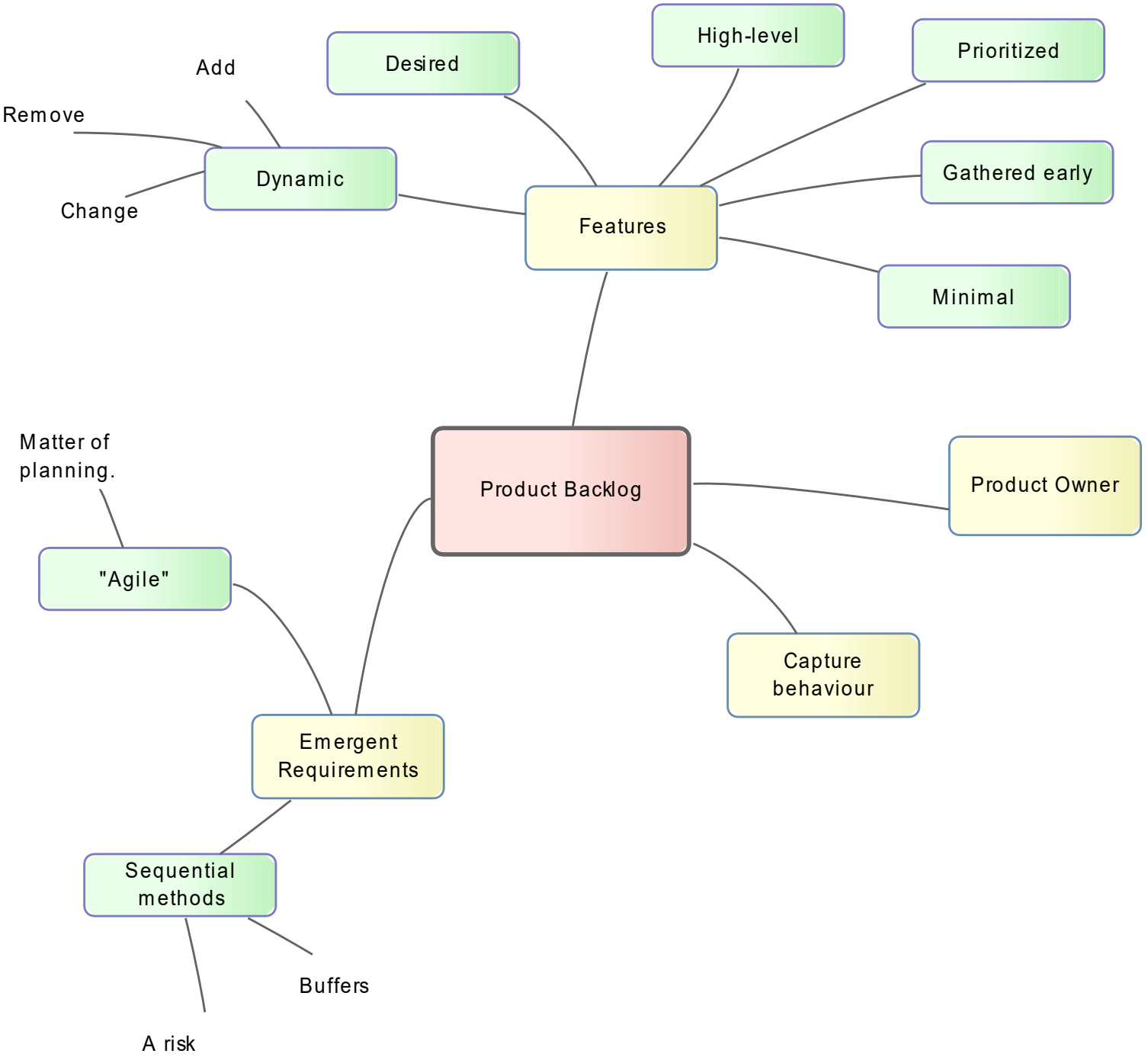
ID:	[Unique ID of this use case]
Title:	[Enter the goal of the use case - preferably as a short, active verb phrase]
Description:	[Describe the goal and context of this use case. This is usually an expanded version of what you entered in the "Title" field.]
Primary Actor:	[A person or a software/hardware system that interacts with your system to achieve the goal of this use case.]
Preconditions:	[Describe the state the system is in before the first event in this use case.]
Postconditions:	[Describe the state the system is in after all the events in this use case have taken place.]
Main Success Scenario:	[Describe the flow of events from preconditions to postconditions, when nothing goes wrong. This is the meat of the use case.]
Extensions:	[Describe all the other scenarios for this use case - including exceptions and error cases.]
Frequency of Use:	[How often will this use case be used?]
Status:	[Development status]
Owner:	[Who owns this use case, in your project team?]
Priority:	[Priority of this use case]

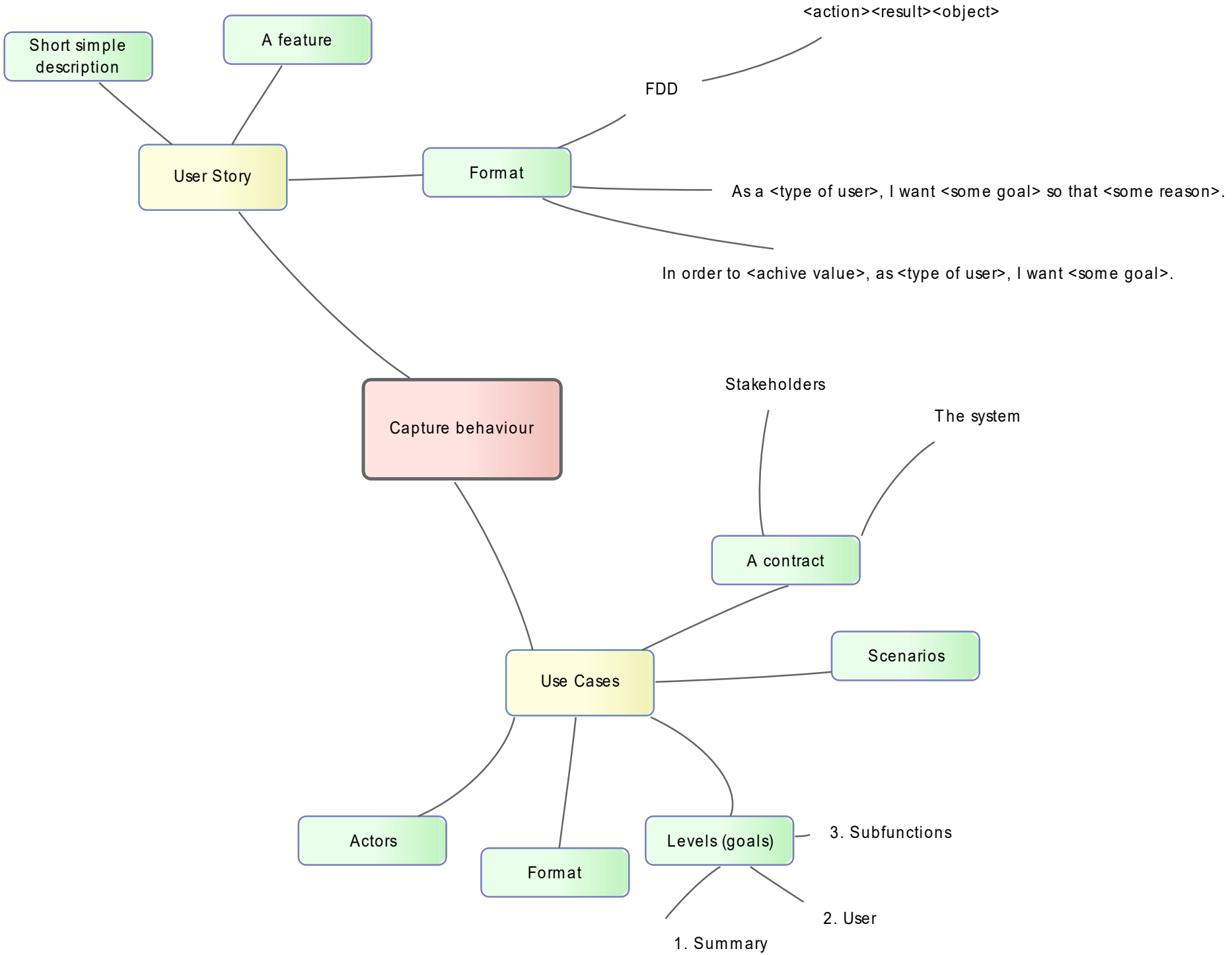
UC Template - Casual

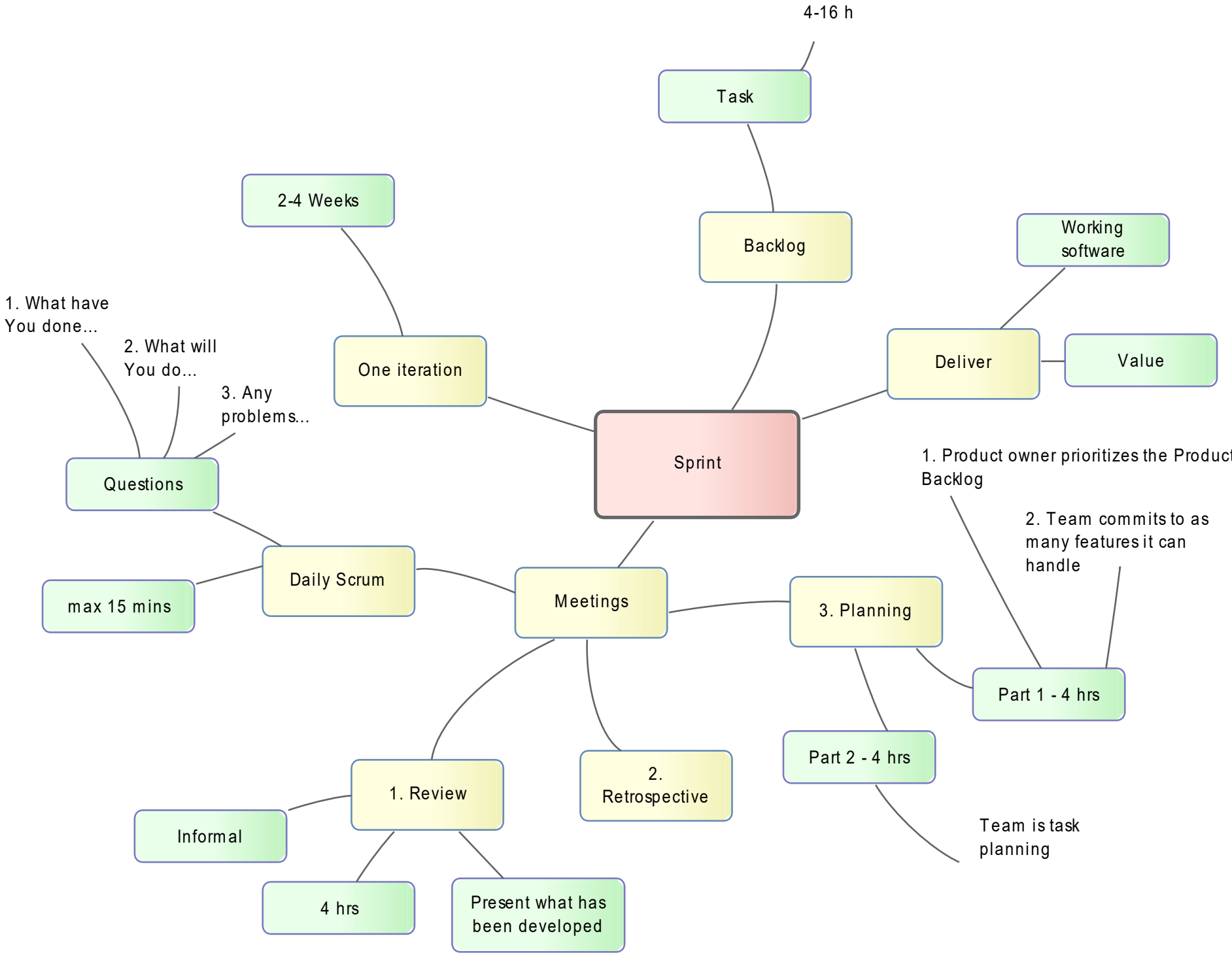
- Free form
  - Title
  - Description
  - (Primary actor
  - Scope
  - Level)

SCRUM


turning a
product backlog
-> increment
the success
Responsible for
Team
Self-managing
Self-organized
Cross-functional
Roles
The product
backlog
Release plans
Product Owner
ROI Objectives
Scrum master
Teaching
Scrum
Implementing
Scrum
Leading/influencing.
Not managing.
Ensure that
everyone
follows Scrum
SCRUM
Rework scheduling
strategy
Iterative
Staging and
scheduling strategy
Incremental
Sprints
Artifacts
Product
backlog
Sprint Backlog


Add
Desired
High-level
Prioritized
Remove
Dynamic
Gathered early
Change
Features
Minimal
Matter of
planning.
Product Backlog
Product Owner
"Agile"
Capture
behaviour
Emergent
Requirements
Sequential
methods
Buffers
A risk


Short simple description
A feature
User Story
<action><result><object>
FDD
Format
As a <type of user>, I want <some goal> so that <some reason>.
In order to <achive value>, as <type of user>, I want <some goal>.
Capture behaviour
Stakeholders
The system
A contract
Use Cases
Scenarios
Actors
Format
Levels (goals)
3. Subfunctions
2. User
1. Summary


4-16 h
Task
2-4 Weeks
Backlog
Working software
1. What have You done...
2. What will You do...
3. Any problems...
One iteration
Deliver
Value
Sprint
Questions
1. Product owner prioritizes the Product Backlog
2. Team commits to as many features it can handle
Daily Scrum
max 15 mins
Meetings
3. Planning
Part 1 - 4 hrs
Part 2 - 4 hrs
1. Review
2. Retrospective
Informal
Team is task planning
4 hrs
Present what has been developed

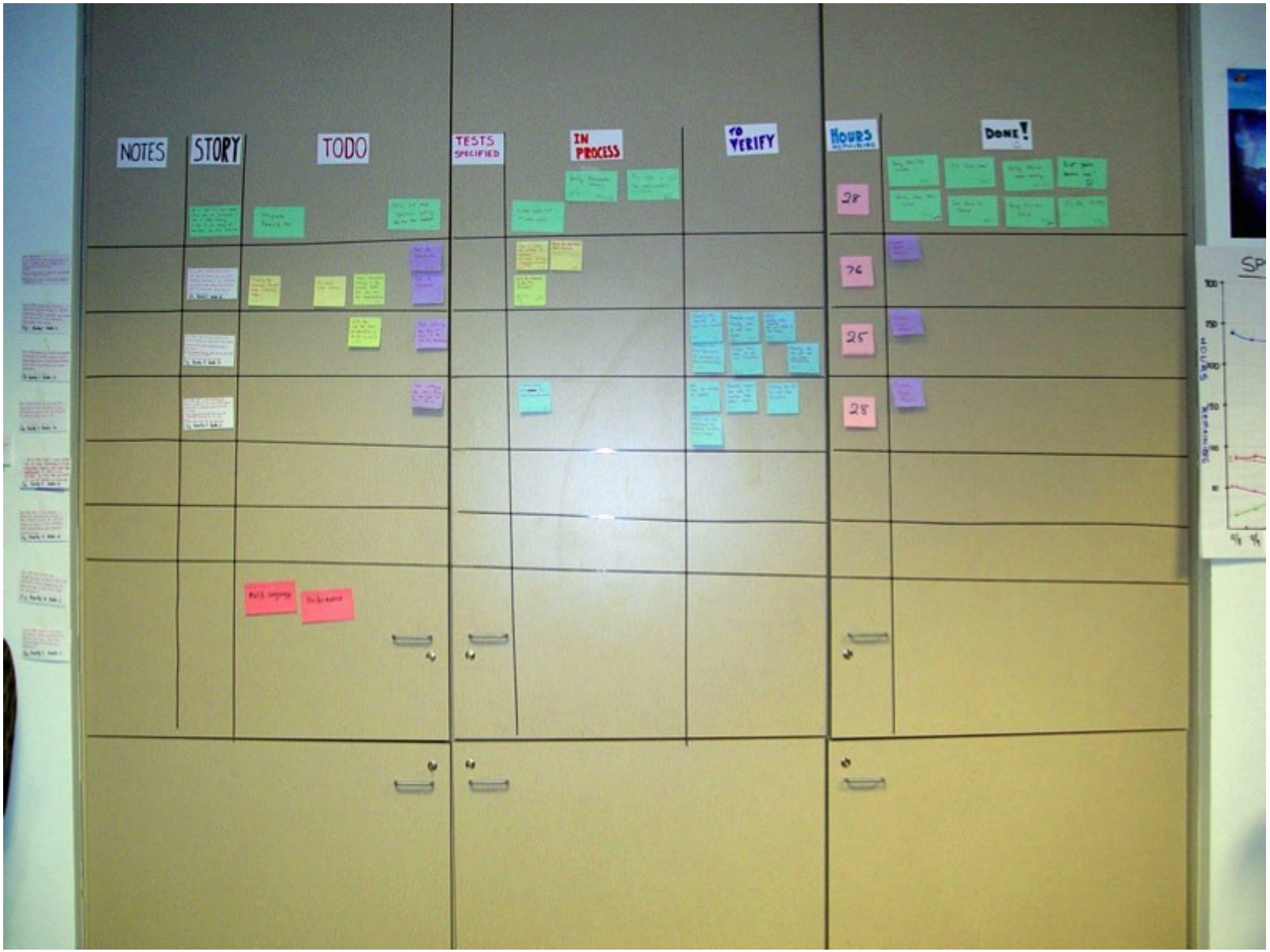
Example Product Backlog

ID	Story	Priority	Status

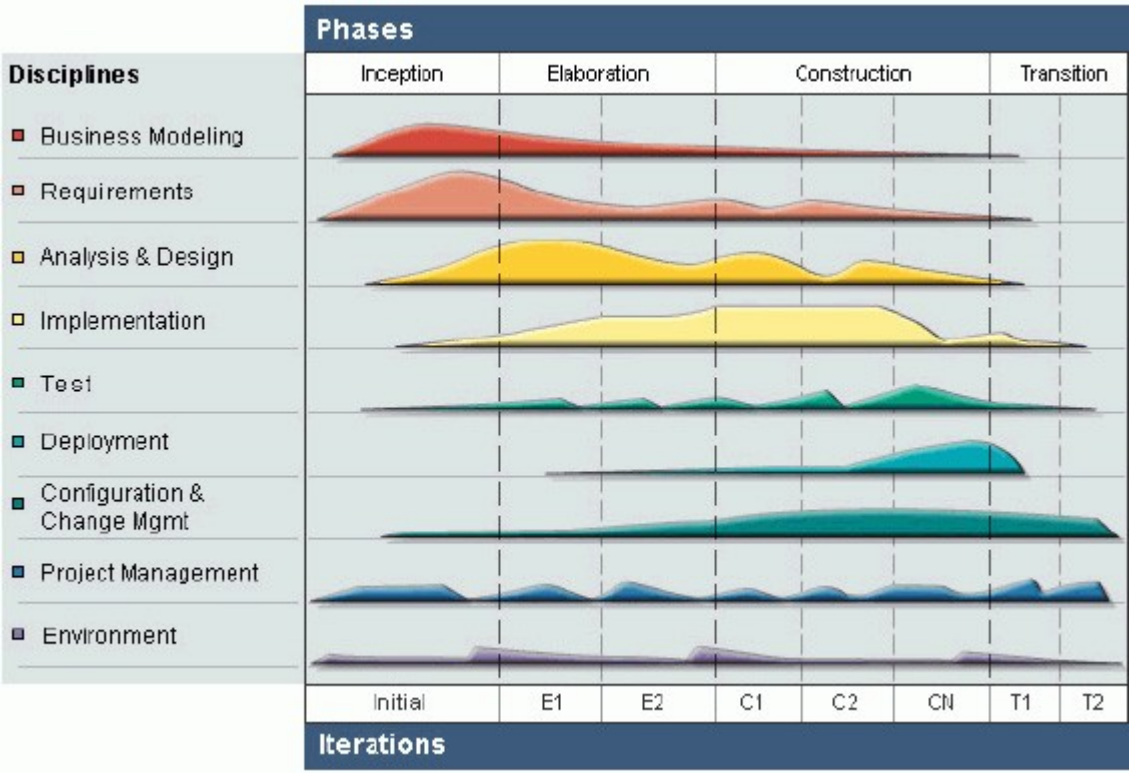
Sprint backlog

Story	Todo	In process	To Verify	Done
I as a ……	Code the .. 1 hrs Code the .. 9 hrs	Code the .. 9 hrs LK		Code the .. 4 hrs NN
I as a ……				

Sprint backlog



UP vs Scrum



- UP is unbalanced
- UP is architecture centric
- Pre-study ?

Method-Fanatic ?

- Nope.
- There is always a context. I use whatever works for the particular problem.
- And I (often) mix...

Why not use a Pre-Study ?

Some Personal Opinions and Experiences

Projects

- A project manager going bezerk
- A psycopath
- Too much workload
- An uncontrollable customer
- A bad contract

Capacity utilisation

4 out 5 days

Never …

ASSUME

That will make an ASS out of U and ME

Add value...

to the Customer!

You are supposed to...

make Your company profitable!

Communicate

If You don't understand, ask again and again and again.

Technical Debt - 1

Is sometimes the only way forward.

Bill payment & due dates!

Technical Debt - 2

Managers and custmers won't remember what You told them.

Technology is seldom...

the problem. People are.

People

Are different!

Brainstorming...

Is (mostly) bad.

Open plan offices...

Is (mostly) bad.

I want my door back!

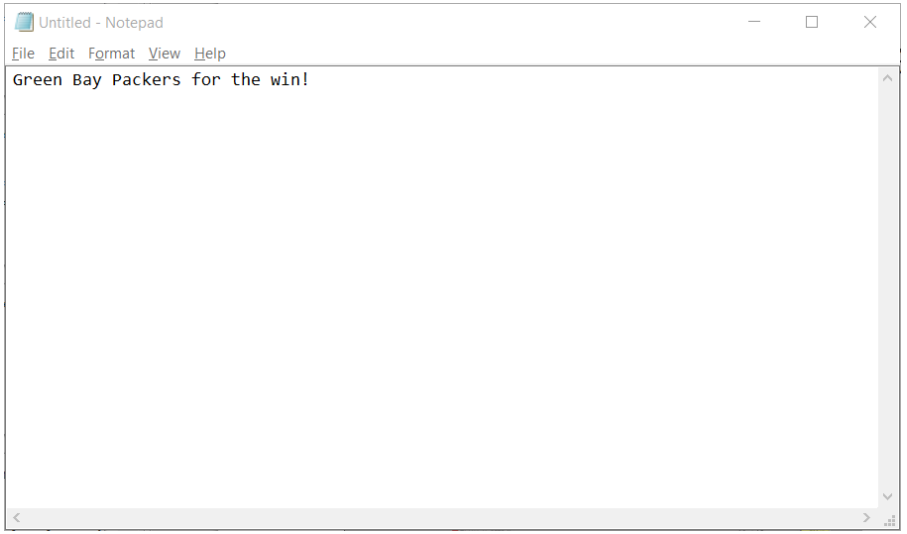
Users

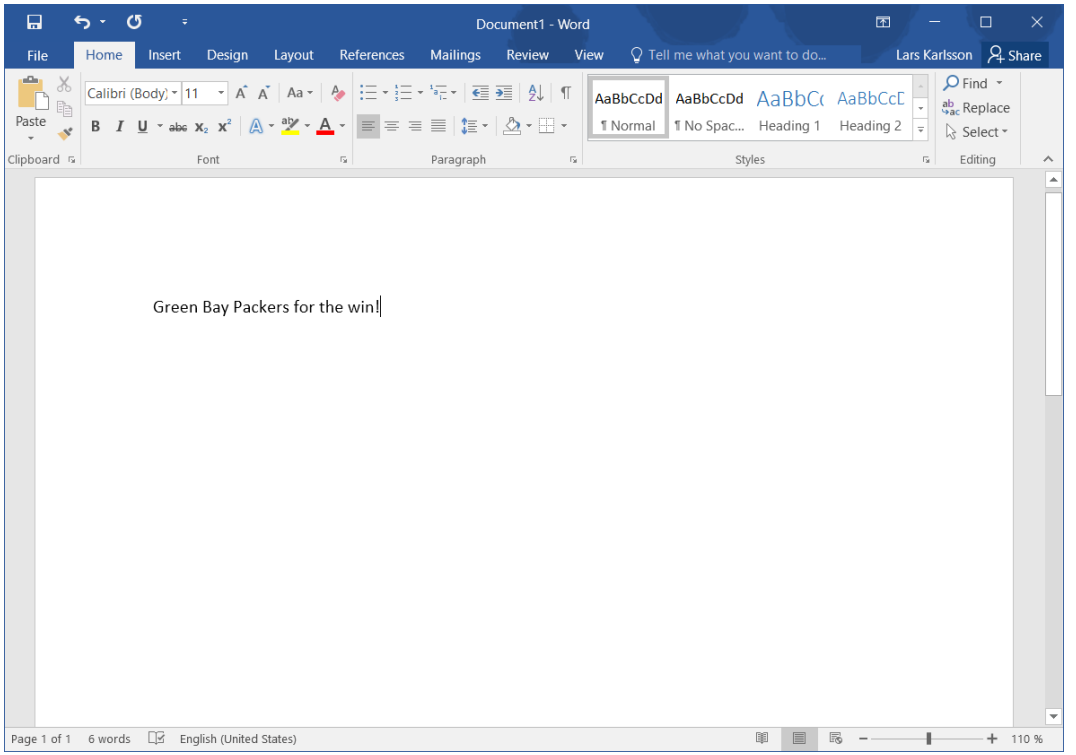
Has lower computer skills than You think.

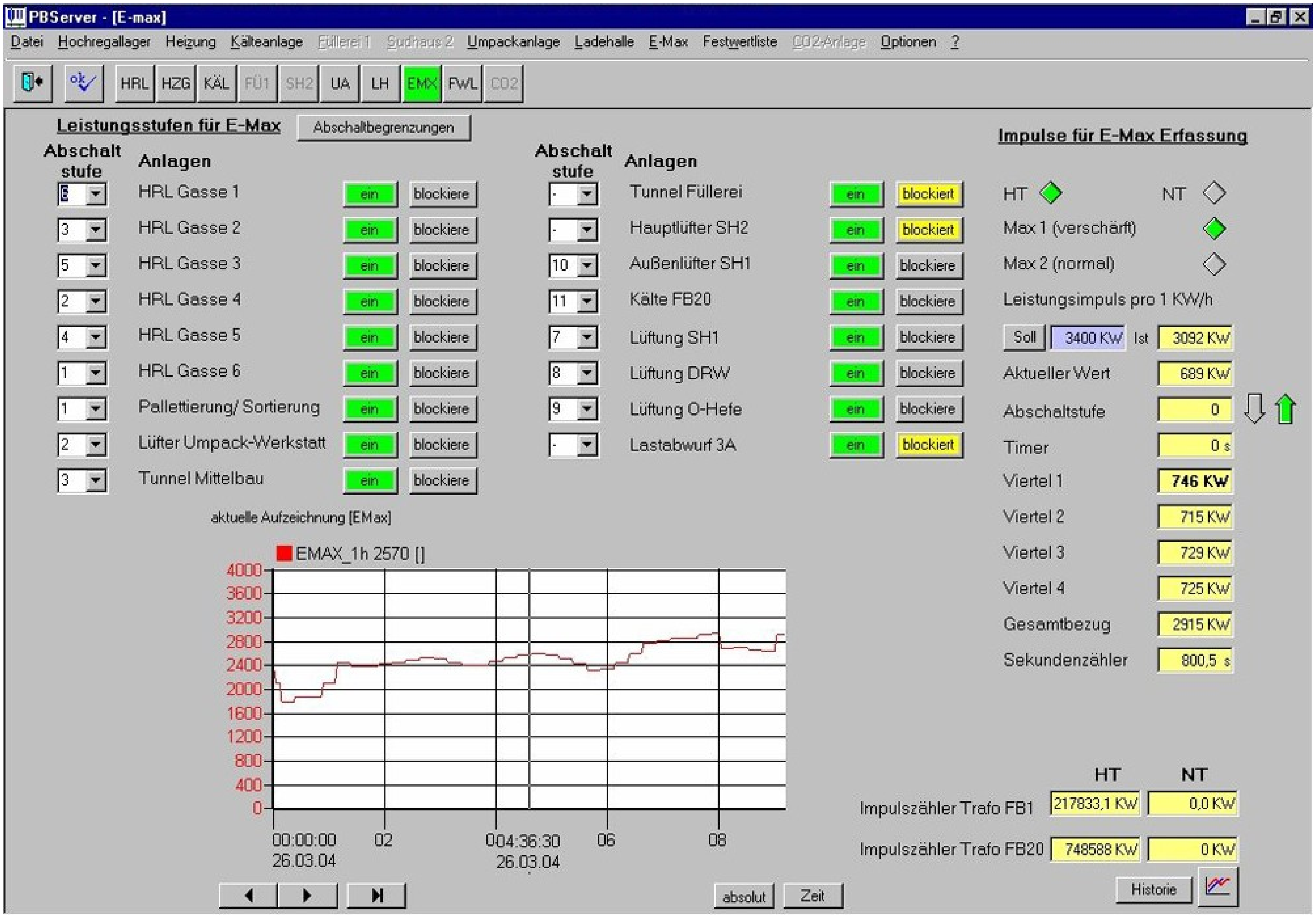
Users in general 1

Wants stuff that is good enough.
(they won't pay You for exellence)

Users in general 2



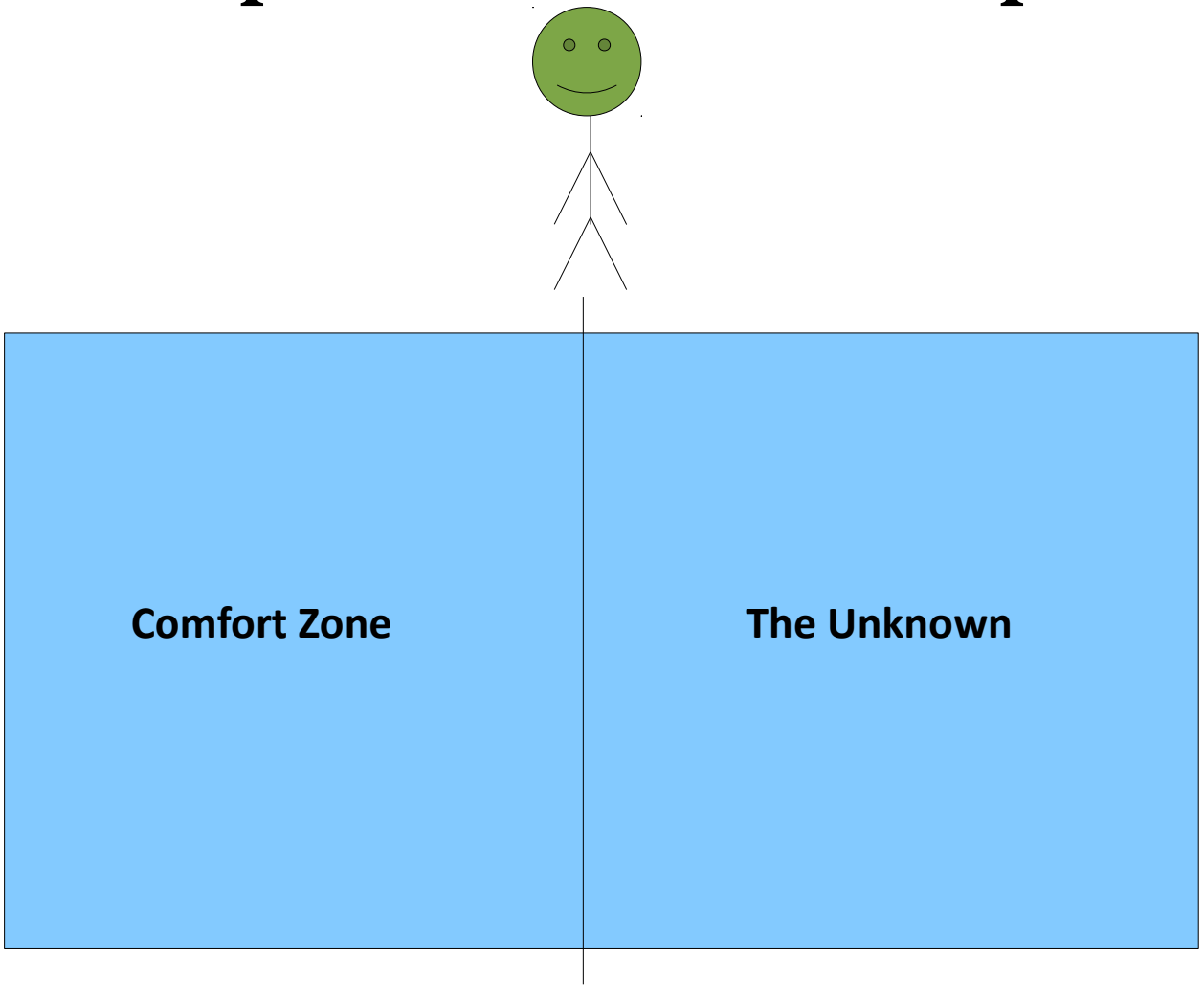



PBServer - [E-max]
Datei Hochregallager Heizung Kälteanlage Füllerei 1 Sudhaus 2 Umpackanlage Ladehalle E-Max Festwertliste CO2-Anlage Optionen ?
HRL HZG KÄL FÜ1 SH2 UA LH EMX FWL CO2
Leistungsstufen für E-Max
Abschaltbegrenzungen
Abschalt stufe
Anlagen
HRL Gasse 1
HRL Gasse 2
HRL Gasse 3
HRL Gasse 4
HRL Gasse 5
HRL Gasse 6
Pallettierung/ Sortierung
Lüfter Umpack-Werkstatt
Tunnel Mittelbau
Tunnel Füllerei
Hauptlüfter SH2
Außenlüfter SH1
Kälte FB20
Lüftung SH1
Lüftung DRW
Lüftung O-Hefe
Lastabwurf 3A
ein
blockiere
blockiert
Impulse für E-Max Erfassung
HT
NT
Max 1 (verschärft)
Max 2 (normal)
Leistungsimpuls pro 1 KW/h
Soll 3400 KW Ist 3092 KW
Aktueller Wert 689 KW
Abschaltstufe 0
Timer 0 s
Viertel 1 746 KW
Viertel 2 715 KW
Viertel 3 729 KW
Viertel 4 725 KW
Gesamtbezug 2915 KW
Sekundenzähler 800,5 s
aktuelle Aufzeichnung [EMax]
EMAX_1h 2570 []
00:00:00 26.03.04
004:36:30 26.03.04
absolut
Zeit
Impulszähler Trafo FB1 217833,1 KW 0,0 KW
Impulszähler Trafo FB20 748588 KW 0 KW
Historie

Are You a...

Manager or a Leader? There is a difference!

Constant personal development



Some useful links

- Managing the development of large software systems. http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf
- Manifest for Agile software development. www.agilemanifesto.org.
- Software engineering method and theory (SEMAT). www.semat.org.
- Agile modeling – effectice practices for modeling and documentation. www.agilemodeling.com.
- Scrum. www.scrum.org.
- What is Scrum? www.mountaingoatsoftware.com/topics/scrum.

Books

- DeMarco, Tom and Timothy Lister. 1999. *Peopleware – Productive Projects and Teams.* Dorset House Publishing.
- Brooks, Frederick. 2010. *The Design of Design.* Addison-Wesley.
- Thomke, Stefan. 2003. *Experimentation Matters.* Harvard Business School Press.
- Highsmith, Jim. 2010. *Agile Project Management.* Addison-Wesley.
- Cockburn, Alistair. 2001.*Writing Effective Use Cases*. Addison-Wesley.
- Schwaber, Ken. 2004. *Agile Project Management With Scrum*. Microsoft Press.
- Cohn, Brian. 2010. *Succeeding With Agile – Software Development Using Scrum*. Addison-Wesley.
- Rosenberg, Doug, Matt Stephens, and Mark Collins-Cope. 2005. *Agile development with ICONIX process – People, Process and Pragmatism*. Apress.
- Arlow, Jim and Ila Neustadt. 2005. *UML2 and the unified process*. Addison-Wesley.

Peopleware

The Flow

- A zone where You are fully concentrated
- >15 min to reach

Environment factor

$$EF = \frac{Uninterrupted\ Hours}{BodyPresent\ Hours}$$