Advanced Feedback Driven Development

What do you do when "unexpected" things happen? Do you ignore it or do you change your plans accordingly? During the last decade we have seen a big shift in the way knowledge workers, software engineers in particular, are supposed to organize their activities. Slack, agility and effectiveness are typical words used when describing the "new way", while plans, metrics and objectives are typical words used to describe the "old way" of working. In this talk we will describe many of the new concepts and discuss how they can improve the effectiveness of individuals, teams and organizations.

slides for a 60 minute session, Cisco, CETG, Bangalore
November 13, 2014

Olve Maudal

Advanced Feedback Driven Development

some random rants about software engineering

What do you do when "unexpected" things happen? Do you ignore it or do you change your plans accordingly? During the last decade we have seen a big shift in the way knowledge workers, software engineers in particular, are supposed to organize their activities. Slack, agility and effectiveness are typical words used when describing the "new way", while plans, metrics and objectives are typical words used to describe the "old way" of working. In this talk we will describe many of the new concepts and discuss how they can improve the effectiveness of individuals, teams and organizations.

slides for a 60 minute session, Cisco, CETG, Bangalore
November 13, 2014

Olve Maudal

- About high-tech product development
- About Agile Principles
- About Lean Thinking
- Some principles of effective product development
- About multi-site development


cisco

T = 360 Myr

About high-tech product development

Few high tech projects are like running
down on a paved road where you can see the ...

... goal in the end of the road.

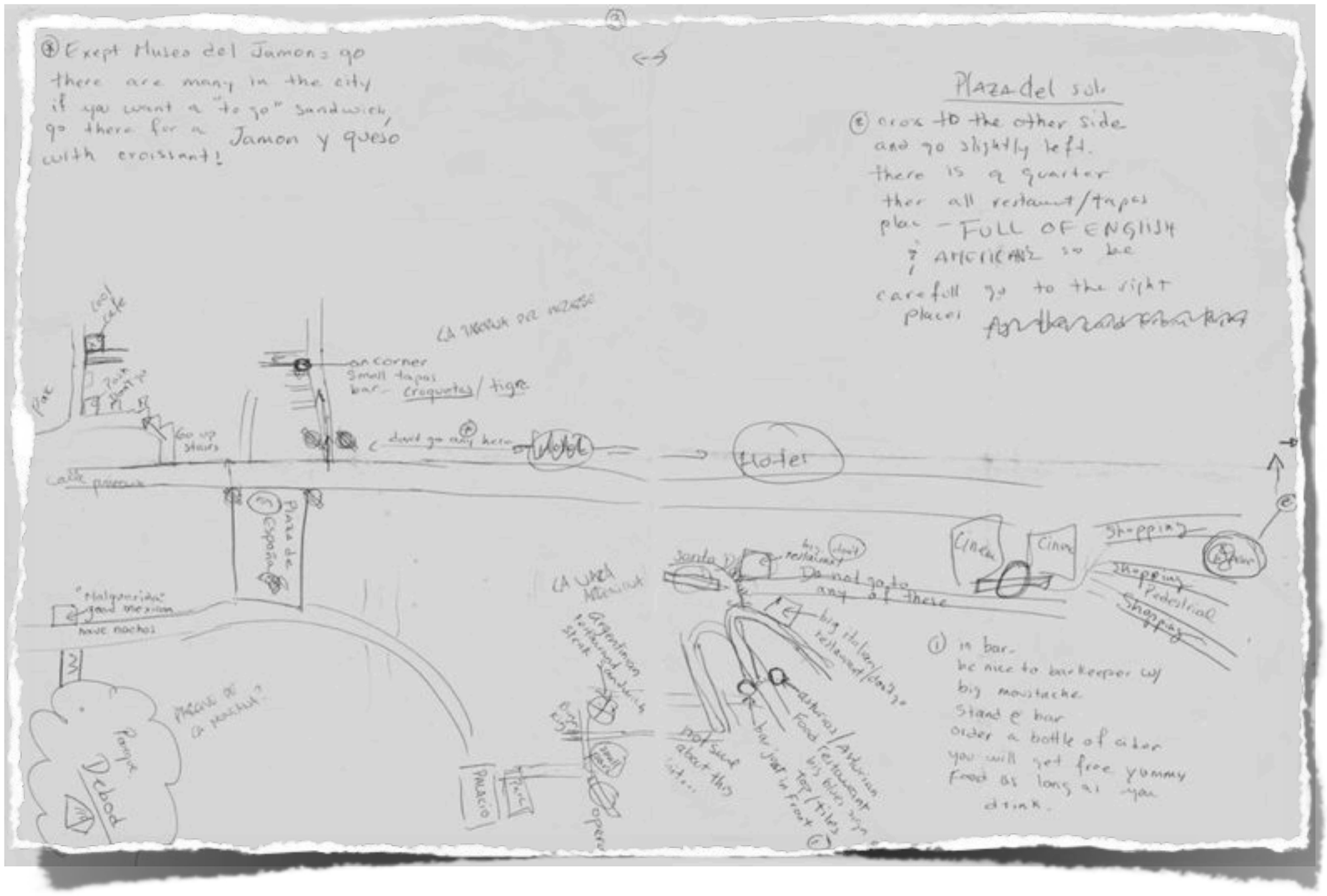
Most projects are more like...

extreme orienteering

in impossible terrain

with a group of people

in the dark

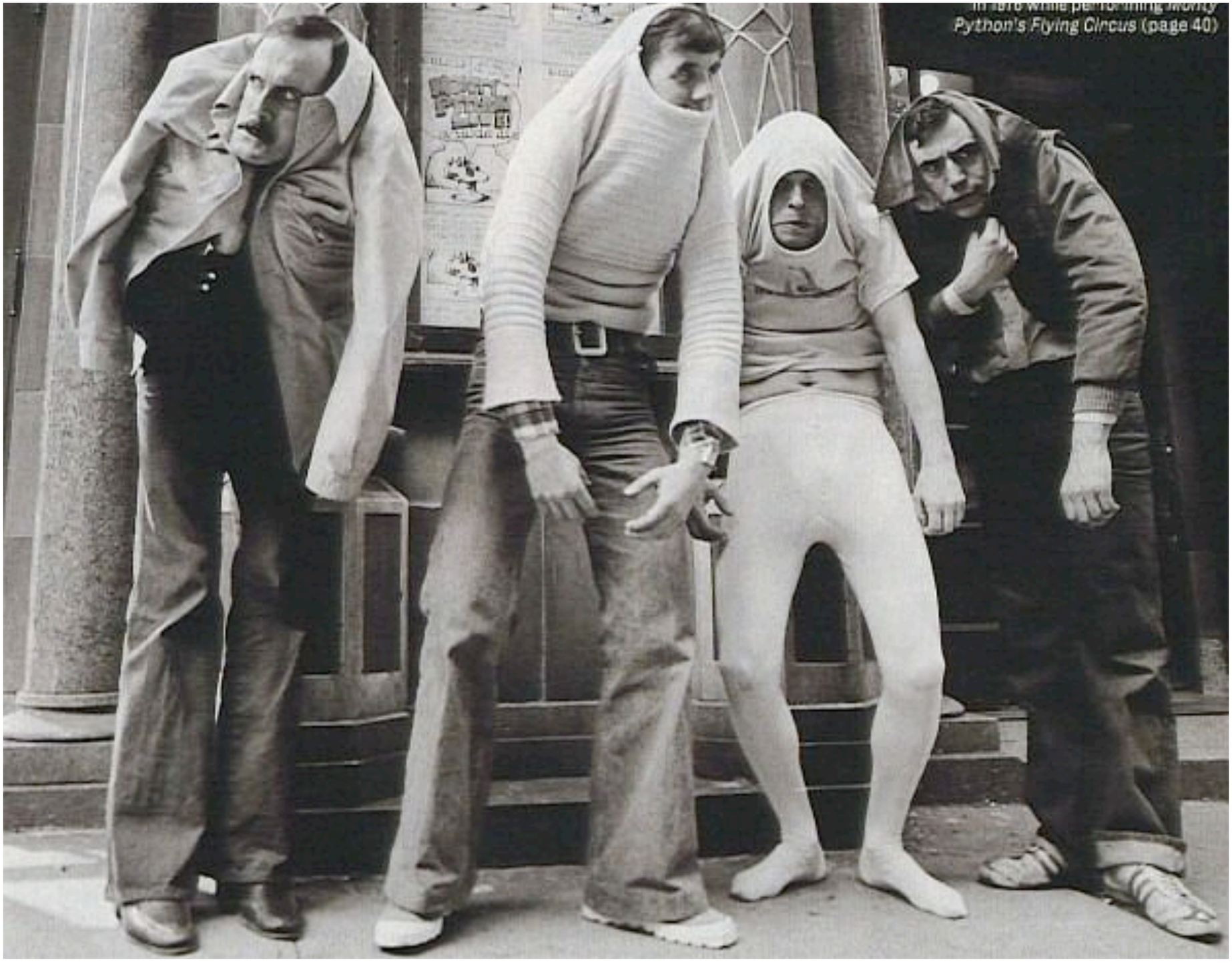


with only a sketchy map as guidance

http://www.youtube.com/watch?v=oetF3UTIwbc

About Agile Principles

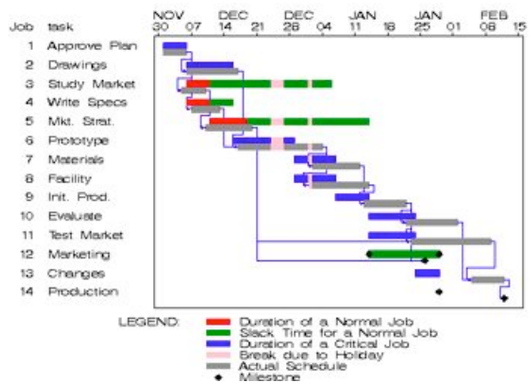
There used to be a time, where we believed that anyone could do software development



after all, it was just about programming a computer...

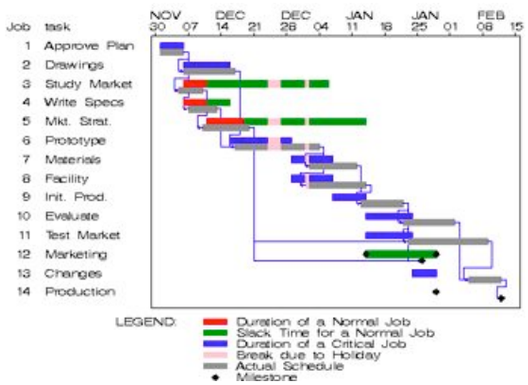
1) get some smart people to analyze the problem

1) get some smart people to analyze the problem



2) create a plan

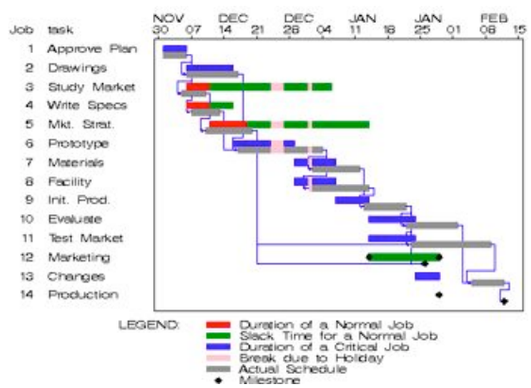
1) get some smart people to analyze the problem



2) create a plan

3) find resources

1) get some smart people to analyze the problem



2) create a plan

3) find resources

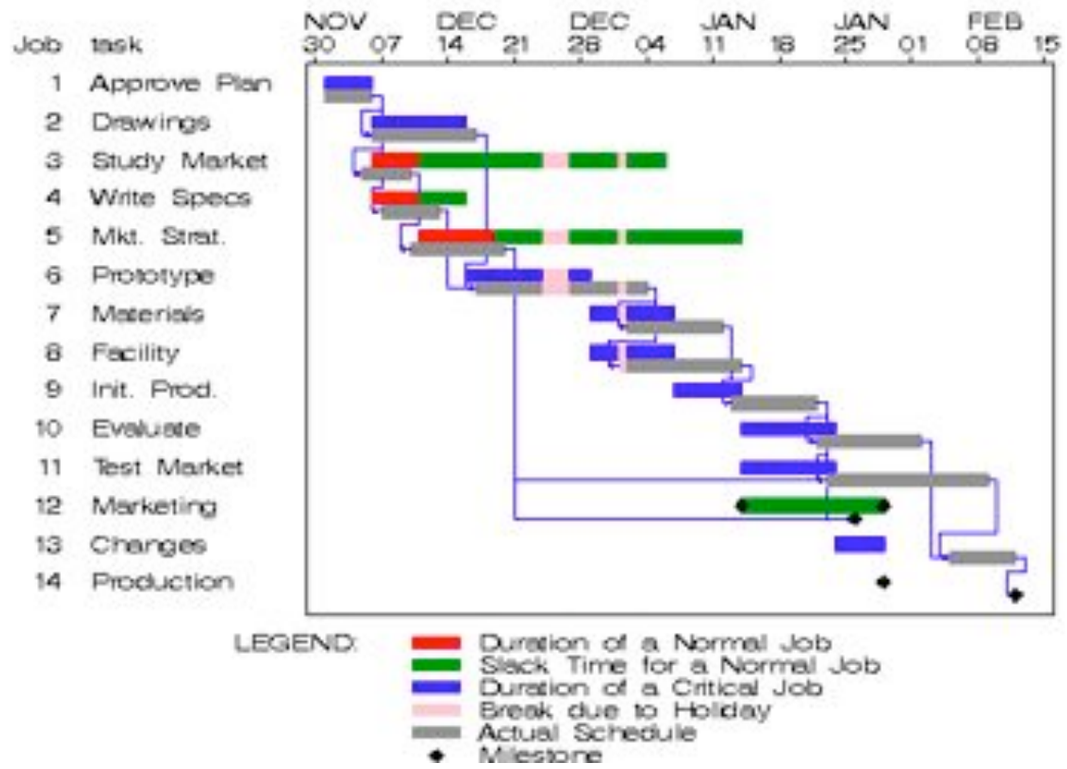
4) execute according to the plan

and when the projects failed

the respons was always:

do more up-front analysis

create a more detailed plan



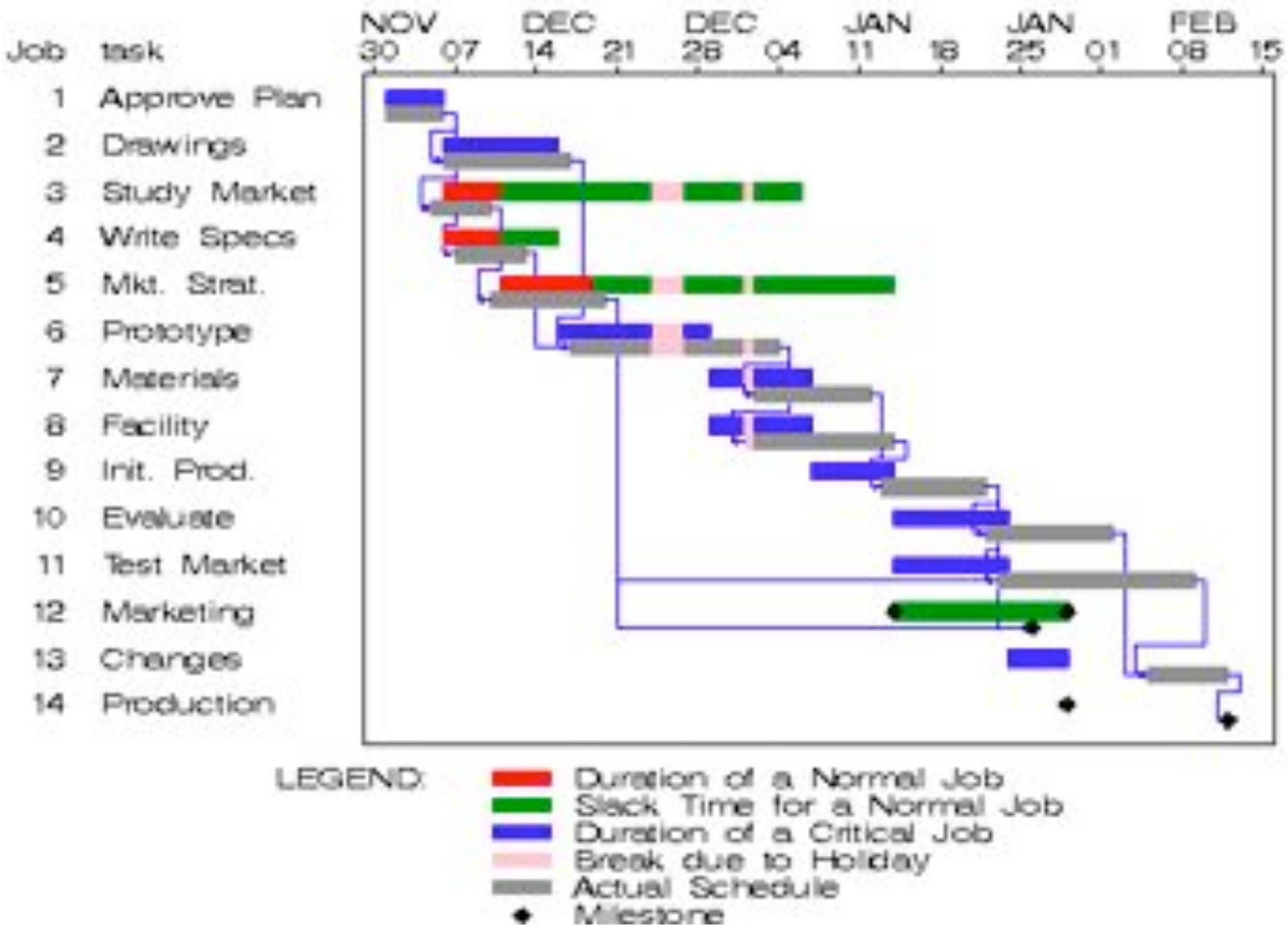
find more resources

and make sure that everyone followed the plan

but the projects still failed

and the respons was, as usual:

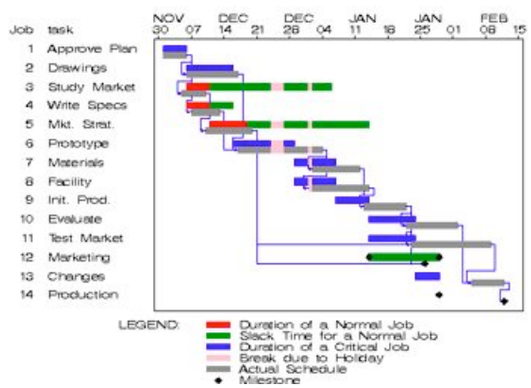

FREE
OR THE UNEMPLOYED
FREE SOUP


Job
task
NOV
DEC
DEC
JAN
JAN
FEB
30
07
14
21
28
04
11
18
25
01
08
15
1 Approve Plan
2 Drawings
3 Study Market
4 Write Specs
5 Mkt. Strat.
6 Prototype
7 Materials
8 Facility
9 Init. Prod.
10 Evaluate
11 Test Market
12 Marketing
13 Changes
14 Production
LEGEND:
Duration of a Normal Job
Slack Time for a Normal Job
Duration of a Critical Job
Break due to Holiday
Actual Schedule
Milestone

but of course...

Dark ages of software development (early 80's to late 90's)

1) get some smart people to analyze the problem



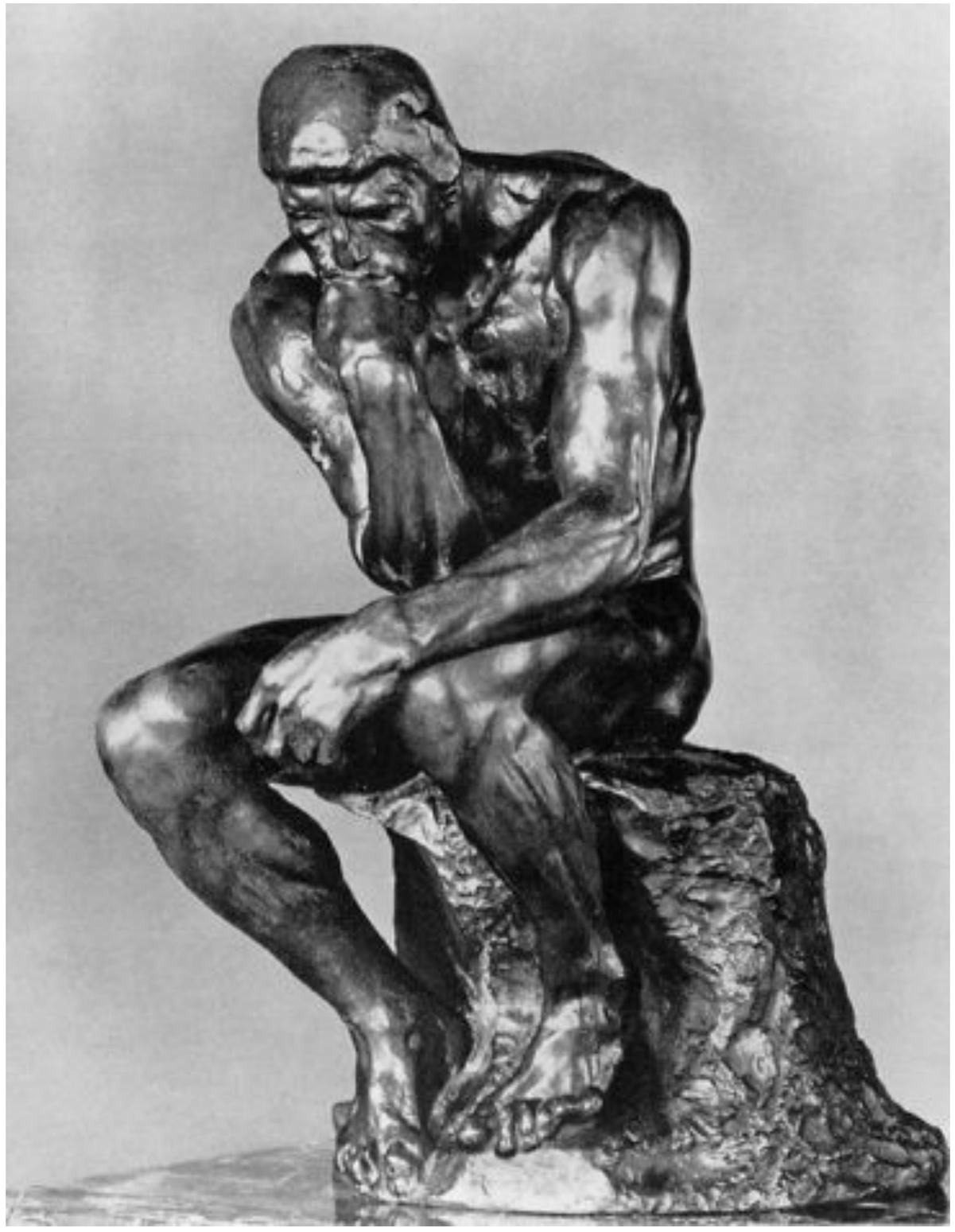
2) create a plan

3) find resources

4) execute according to the plan

We had only discovered a fancy way of playing the “scabby queen” game, also known as the “Old Maid” or “Svarte Per”, always try to “save your ass” by delegating responsibility to someone else.

There must be a better way...



The Agile Manifesto (2001)

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.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

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

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

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

The agile manifesto started a huge awakening process in the software industry...

(picture from the 1990 film Awakenings)

but we also see that similar realization is reaching other disciplines now.

(picture from the 1990 film Awakenings)

processes and tools
comprehensive documentation
contract negotiation
following a plan

processes and tools
comprehensive documentation
contract negotiation
following a plan

Individuals and interactions
Working solutions
Customer collaboration
Responding to change

Individuals and interactions
Working solutions
Customer collaboration
Responding to change

Kanban

Roles

Burndown charts

Sprints

Review meetings

Planning poker

Basecamp

Standup meetings

Retrospectives

Rally

Kanban

Sashimi

CONWIP

Backlogs

Jira

Team

Scrum of Scrums

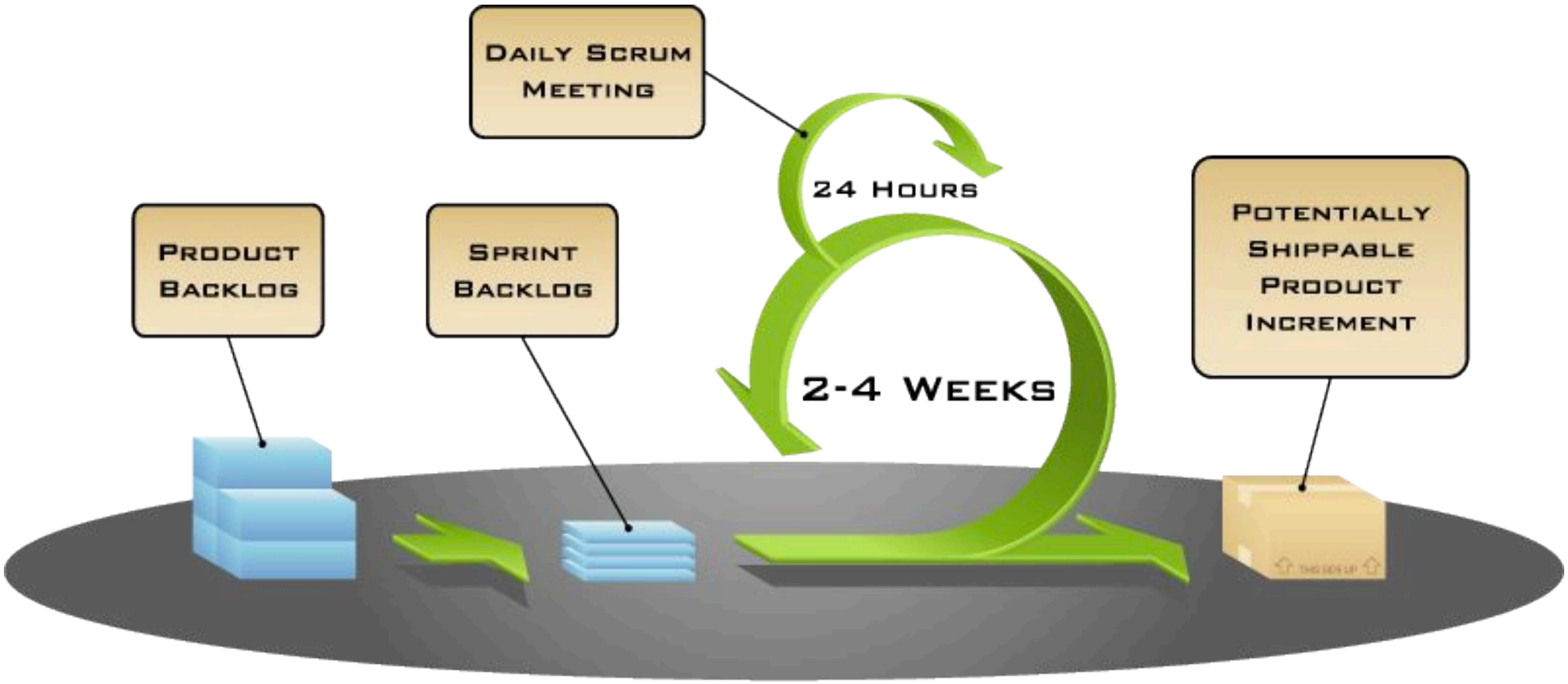
Scrum

Product owner

DoD

Value Stream Mapping

... and much more



Principles behind the Agile Manifesto

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Seven Enemies of Agile

Plans
Commitments
Pressure
Objectives
Documentation
Inspection
Procedures

Plan

Everyone has a plan 'till they get punched in the mouth.
Mike Tyson

Commitments

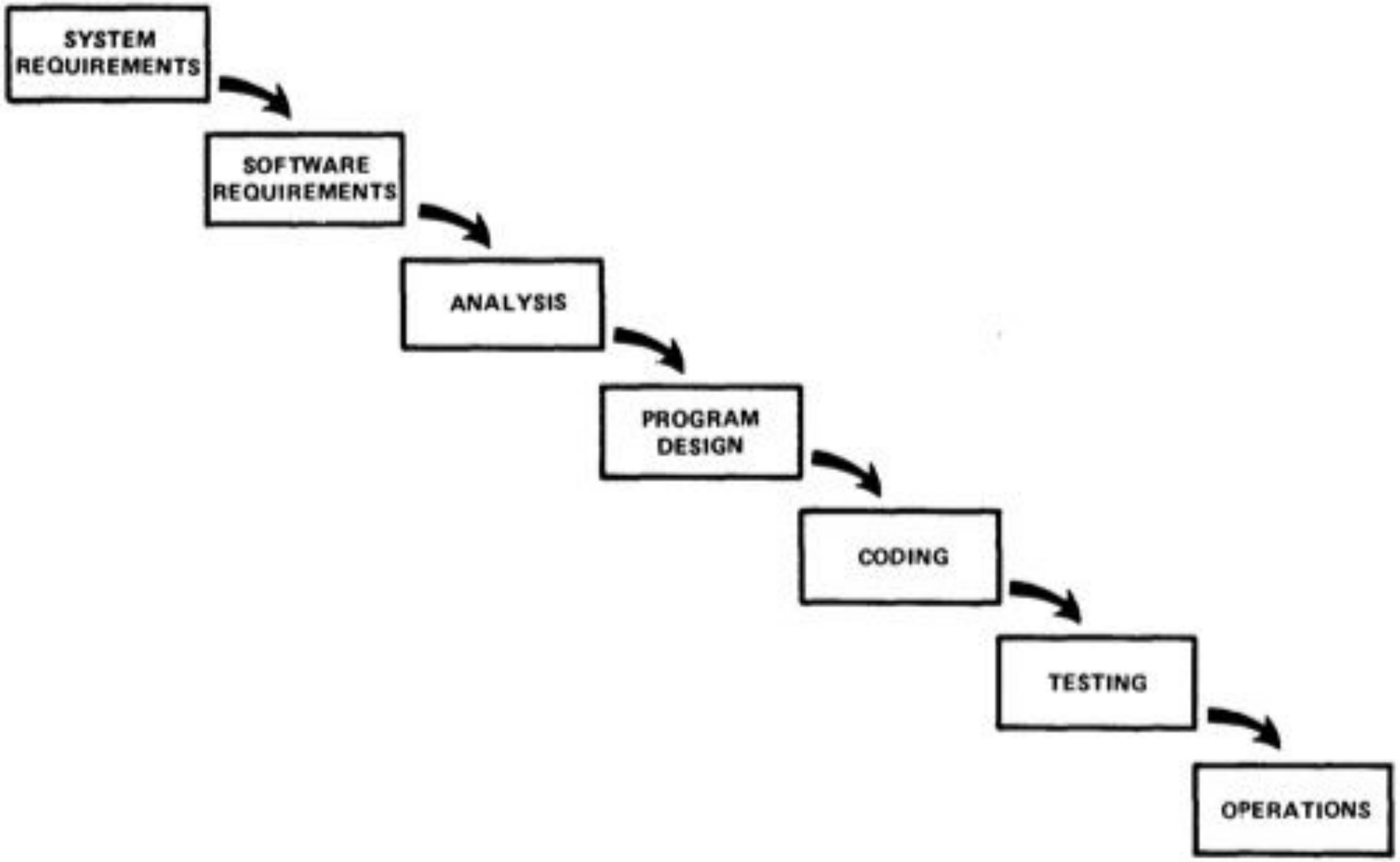
Pressure

Objectives

Documentation

Inspection

Procedures


SYSTEM REQUIREMENTS
SOFTWARE REQUIREMENTS
ANALYSIS
PROGRAM DESIGN
CODING
TESTING
OPERATIONS

Seven Friends of Agile

Seven Friends of Agile

Planning

Seven Friends of Agile

Planning
Collaboration

Seven Friends of Agile

Planning
Collaboration
Slack

Seven Friends of Agile

Planning
Collaboration
Slack
Direction

Seven Friends of Agile

Planning
Collaboration
Slack
Direction
Communication

Seven Friends of Agile

Planning
Collaboration
Slack
Direction
Communication
Reflection

Seven Friends of Agile

Planning
Collaboration
Slack
Direction
Communication
Reflection
Principles

Plans

Planning

Plans

Commitments

Planning

Collaboration

Plans

Commitments

Pressure

Planning

Collaboration

Slack

Plans

Commitments

Pressure

Objectives

Planning

Collaboration

Slack

Direction

Plans

Commitments

Pressure

Objectives

Documentation

Planning

Collaboration

Slack

Direction

Communication

Plans

Commitments

Pressure

Objectives

Documentation

Inspection

Planning

Collaboration

Slack

Direction

Communication

Reflection

Plans

Commitments

Pressure

Objectives

Documentation

Inspection

Procedures

Planning

Collaboration

Slack

Direction

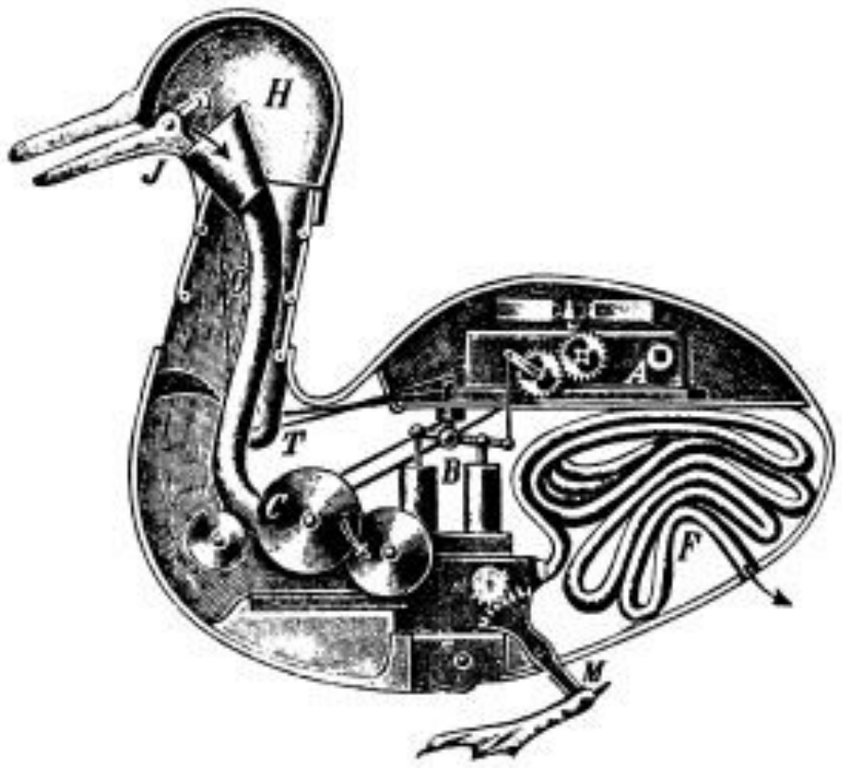
Communication

Reflection

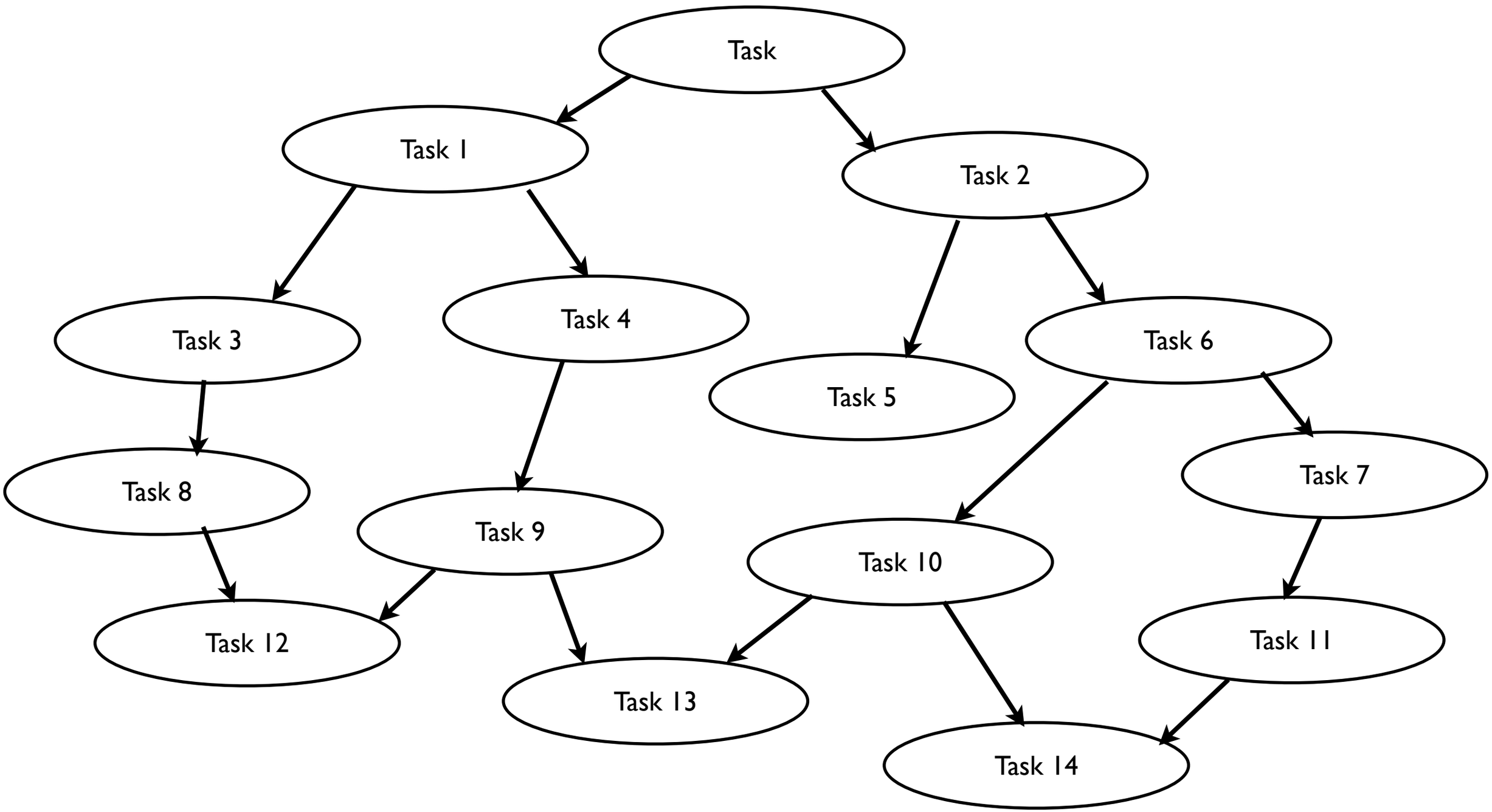
Principles

About Lean Thinking (Demingism)

Reductionism is a philosophical position that a complex system is nothing but the sum of its parts, and that an account of it can be reduced to accounts of individual constituents.



Divide and Conquer



Systems thinking is the process of understanding how things influence one another within a whole

Reductionism vs Systems thinking

Reductionism is a philosophical position that a complex system is nothing but the sum of its parts, and that an account of it can be reduced to accounts of individual constituents.

Systems thinking is the process of understanding how things influence one another within a whole

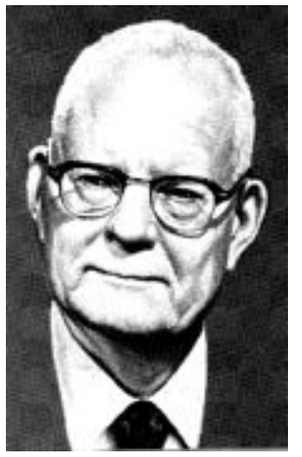
Reductionism vs Systems thinking

Reductionism is a philosophical position that a complex system is nothing but the sum of its parts, and that an account of it can be reduced to accounts of individual constituents.

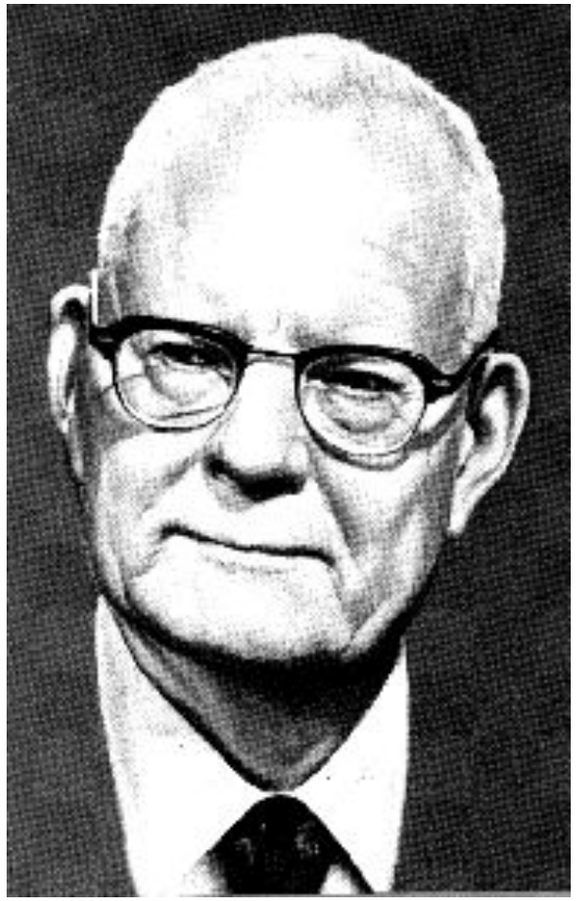
Systems thinking is the process of understanding how things influence one another within a whole

(aka, Taylorism vs Demingism)

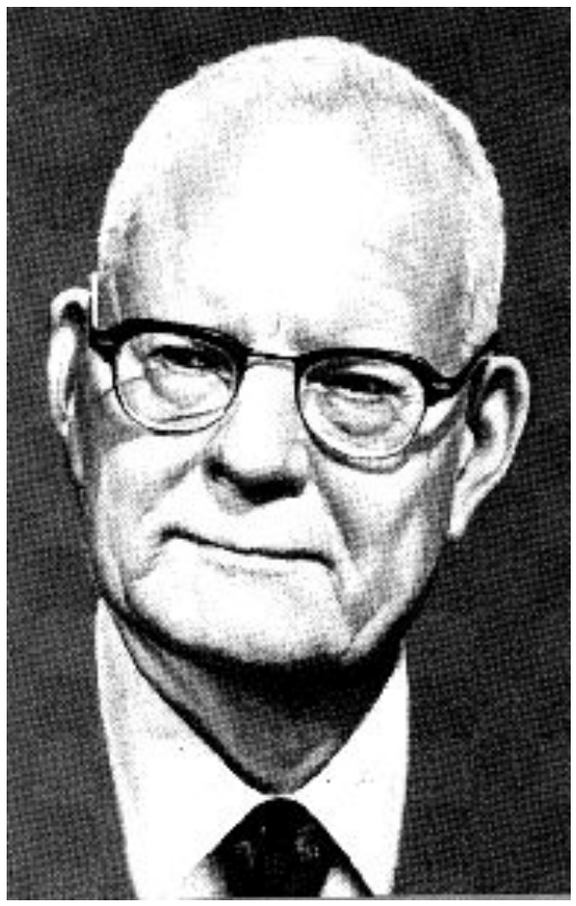
Frederick Winslow Taylor (1856-1915)



W. Edwards Deming (1900-1993)



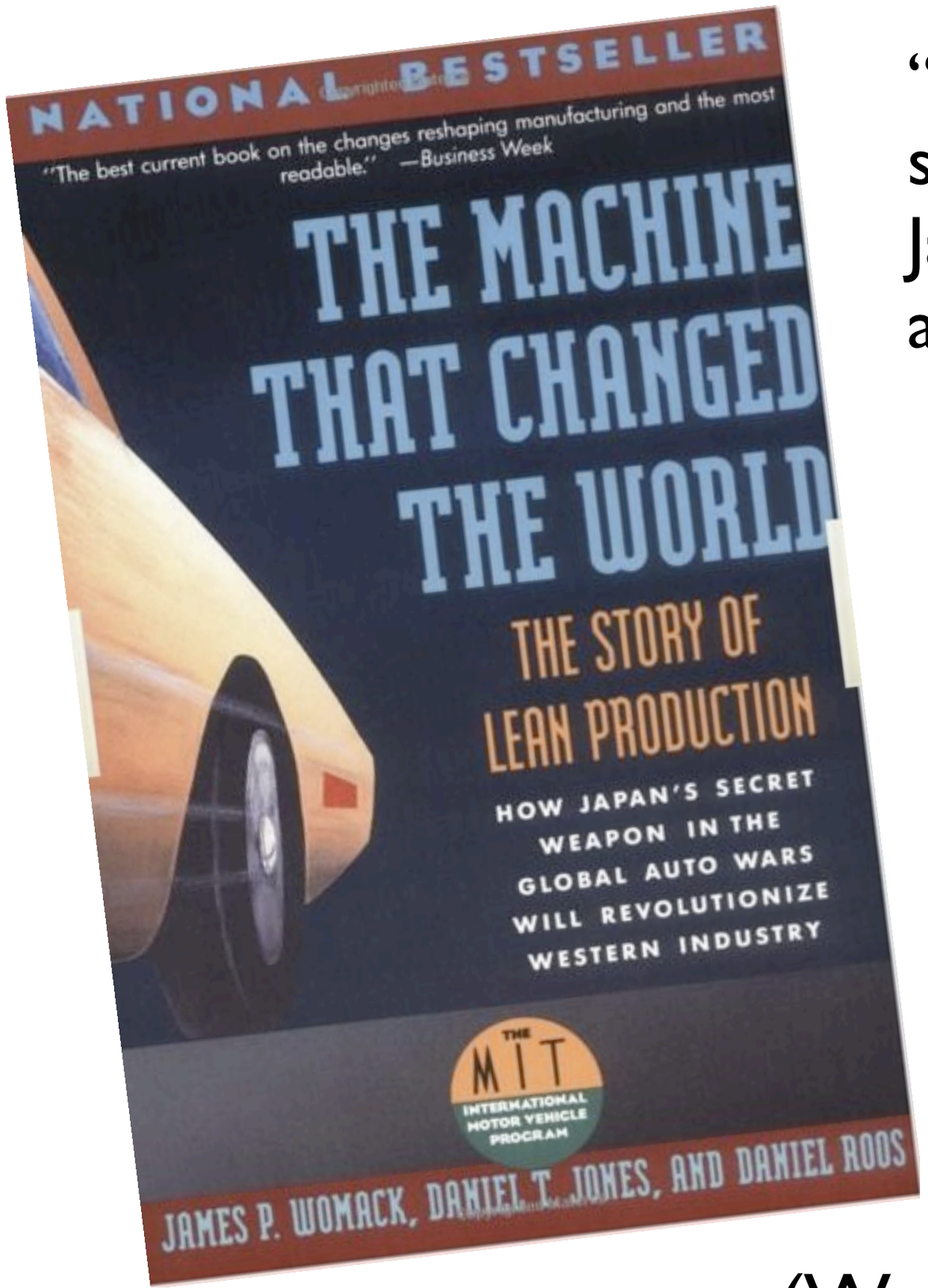
(W. Edwards Deming 1900-1993)



(W. Edwards Deming 1900-1993)

"The striking thing one first notices in the main lobby [in Toyotas HQ] is larger than life pictures of three individuals. One is of Toyota's founder, another of the same size is of Toyota's current chairman, and a third, much larger picture, is of W. Edwards Deming.

"Dr. Deming: The American Who Taught the Japanese About Quality"
(Aquayo, 1991)



"Japanese companies are sweeping the world, and the Japanese auto industry soars above the competition."

(Womack, Jones, Roos, 1990)

(Demingism)

(Taylorism)

Systems thinking is the process of understanding how things influence one another within a whole

Reductionism is a philosophical position that a complex system is nothing but the sum of its parts, and that an account of it can be reduced to accounts of individual constituents.

Demings fourteen key principles for management

Create constancy of purpose toward improvement of product and service, with the aim to become competitive and stay in business, and to provide jobs.

Adopt the new philosophy. We are in a new economic age. Western management must awaken to the challenge, must learn their responsibilities, and take on leadership for change.

Cease dependence on inspection to achieve quality. Eliminate the need for massive inspection by building quality into the product in the first place.

End the practice of awarding business on the basis of price tag. Instead, minimize total cost. **Move towards a single supplier** for any one item, on a long-term relationship of loyalty and trust.

Improve constantly and forever the system of production and service, to improve quality and productivity, and thus constantly decrease costs.

Institute **training** on the job.

Institute leadership (see Point 12 and Ch. 8 of "Out of the Crisis"). The aim of supervision should be to help people and machines and gadgets to do a better job. Supervision of management is in need of overhaul, as well as supervision of production workers.

Drive out fear, so that everyone may work effectively for the company. (See Ch. 3 of "Out of the Crisis")

Break down barriers between departments. **People in research, design, sales, and production must work as a team**, to foresee problems of production and in use that may be encountered with the product or service.

Eliminate slogans, exhortations, and targets for the work force asking for zero defects and new levels of productivity. Such exhortations only create adversarial relationships, as the bulk of the causes of low quality and low productivity belong to the system and thus lie beyond the power of the work force.

a. **Eliminate work standards** (quotas) on the factory floor. Substitute leadership.
b. **Eliminate management by objective**. Eliminate management by numbers, numerical goals. Substitute leadership.

a. **Remove barriers that rob the hourly worker of his right to pride of workmanship**. The responsibility of supervisors must be changed from sheer numbers to quality.
b. Remove barriers that rob people in management and in engineering of their right to pride of workmanship. This means, inter alia," abolishment of the annual or merit rating and of management by objective (See Ch. 3 of "Out of the Crisis").

Institute a vigorous program of education and self-improvement.

Put everybody in the company to work to accomplish the transformation. The transformation is everybody's job.

http://en.wikipedia.org/wiki/W._Edwards_Deming

A subset of Systems Thinking by Deming:

- Cease dependence on inspection to achieve quality
- Move towards single suppliers and establish long-term relationships
- Drive out fear
- Break down barriers between departments
- Eliminate work standards
- Eliminate management by objective
- Remove barriers that rob people their right to pride of workmanship
- Institute a vigorous program of education and self-improvement

A subset of Systems Thinking by Deming:

- Cease dependence on inspection to achieve quality
- Move towards single suppliers and establish long-term relationships
- Drive out fear
- Break down barriers between departments
- Eliminate work standards
- Eliminate management by objective
- Remove barriers that rob people their right to pride of workmanship
- Institute a vigorous program of education and self-improvement

A subset of Systems Thinking by Deming:

- Cease dependence on inspection to achieve quality
- Move towards single suppliers and establish long-term relationships
- Drive out fear
- Break down barriers between departments
- Eliminate work standards
- Eliminate management by objective
- Remove barriers that rob people their right to pride of workmanship
- Institute a vigorous program of education and self-improvement

A subset of Systems Thinking by Deming:

- Cease dependence on inspection to achieve quality
- Move towards single suppliers and establish long-term relationships
- Drive out fear
- Break down barriers between departments
- Eliminate work standards
- Eliminate management by objective
- Remove barriers that rob people their right to pride of workmanship
- Institute a vigorous program of education and self-improvement

A subset of Systems Thinking by Deming:

- Cease dependence on inspection to achieve quality
- Move towards single suppliers and establish long-term relationships
- Drive out fear
- Break down barriers between departments
- Eliminate work standards
- Eliminate management by objective
- Remove barriers that rob people their right to pride of workmanship
- Institute a vigorous program of education and self-improvement

A subset of Systems Thinking by Deming:

- Cease dependence on inspection to achieve quality
- Move towards single suppliers and establish long-term relationships
- Drive out fear
- Break down barriers between departments
- Eliminate work standards
- Eliminate management by objective
- Remove barriers that rob people their right to pride of workmanship
- Institute a vigorous program of education and self-improvement

A subset of Systems Thinking by Deming:

- Cease dependence on inspection to achieve quality
- Move towards single suppliers and establish long-term relationships
- Drive out fear
- Break down barriers between departments
- Eliminate work standards
- Eliminate management by objective
- Remove barriers that rob people their right to pride of workmanship
- Institute a vigorous program of education and self-improvement

Deming states unequivocally that merit reviews, by whatever name, including management by objectives, are the single most destructive force in American management today.

Deming states unequivocally that merit reviews, by whatever name, including management by objectives, are the single most destructive force in American management today.

A subset of Systems Thinking by Deming:

- Cease dependence on inspection to achieve quality
- Move towards single suppliers and establish long-term relationships
- Drive out fear
- Break down barriers between departments
- Eliminate work standards
- Eliminate management by objective
- Remove barriers that rob people their right to pride of workmanship
- Institute a vigorous program of education and self-improvement

A subset of Systems Thinking by Deming:

- Cease dependence on inspection to achieve quality
- Move towards single suppliers and establish long-term relationships
- Drive out fear
- Break down barriers between departments
- Eliminate work standards
- Eliminate management by objective
- Remove barriers that rob people their right to pride of workmanship
- Institute a vigorous program of education and self-improvement

Some principles of effective product development

Embrace chaos

Embrace chaos

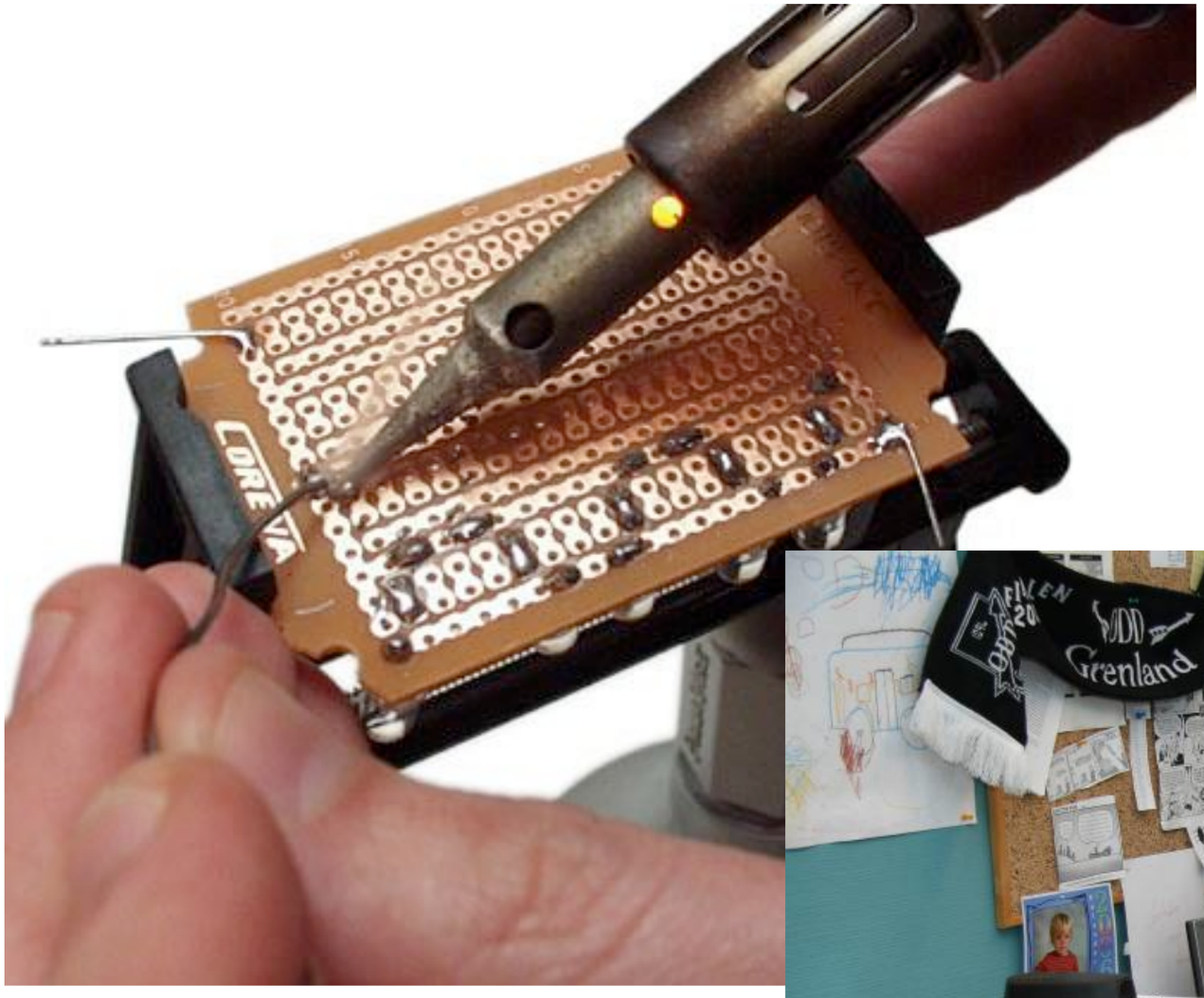
Break the rules

Break the rules

Break the rules

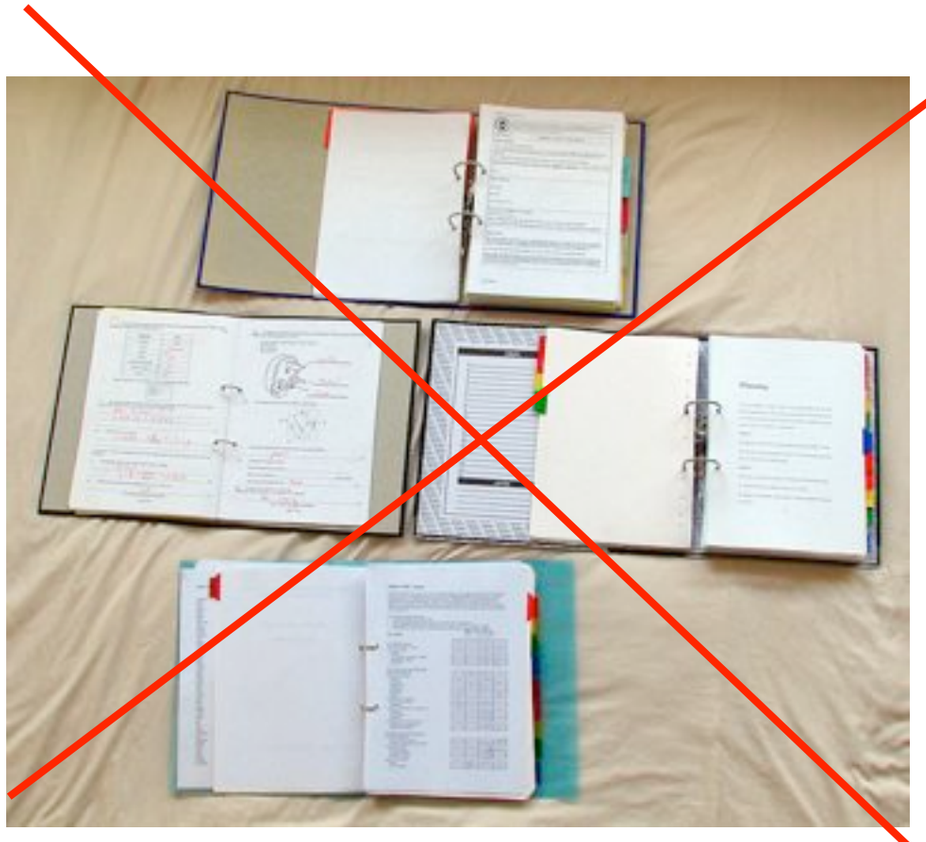
Respect doers (and create an autonomous organization)

Respect doers (and create an autonomous organization)



Focus on communication (over documentation)

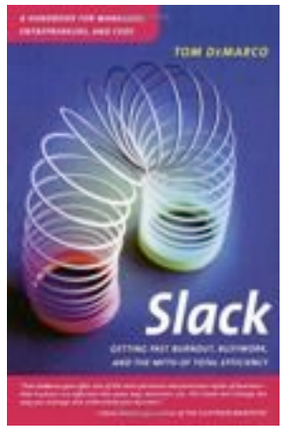
Focus on communication (over documentation)



Introduce slack

Introduce slack

If your company's goal is to become fast, responsive, and agile, more efficiency is not the answer--you need more ***slack****. (Tom DeMarco)*



Beware the observer effect

Beware the observer effect

Constrain your innovation

Constrain your innovation

Reward courage (and failures)

Reward courage (and failures)

Reward courage (and failures)

Focus on the whole product

Focus on the whole product

system thinking vs reductionism

Delay decisions (but do continuous planning)

Delay decisions (but do continuous planning)

Plans are of little importance, but planning is essential – Winston Churchill
Plans are nothing; planning is everything. – Dwight D. Eisenhower
No battle plan survives contact with the enemy. – Helmuth von Moltke the Elder

Delay decisions (but do continuous planning)

Plans are of little importance, but planning is essential – Winston Churchill
Plans are nothing; planning is everything. – Dwight D. Eisenhower
No battle plan survives contact with the enemy. – Helmuth von Moltke the Elder

Everyone has a plan 'till they get punched in the mouth. – Mike Tyson

Aim for approximately right rather than accurately wrong

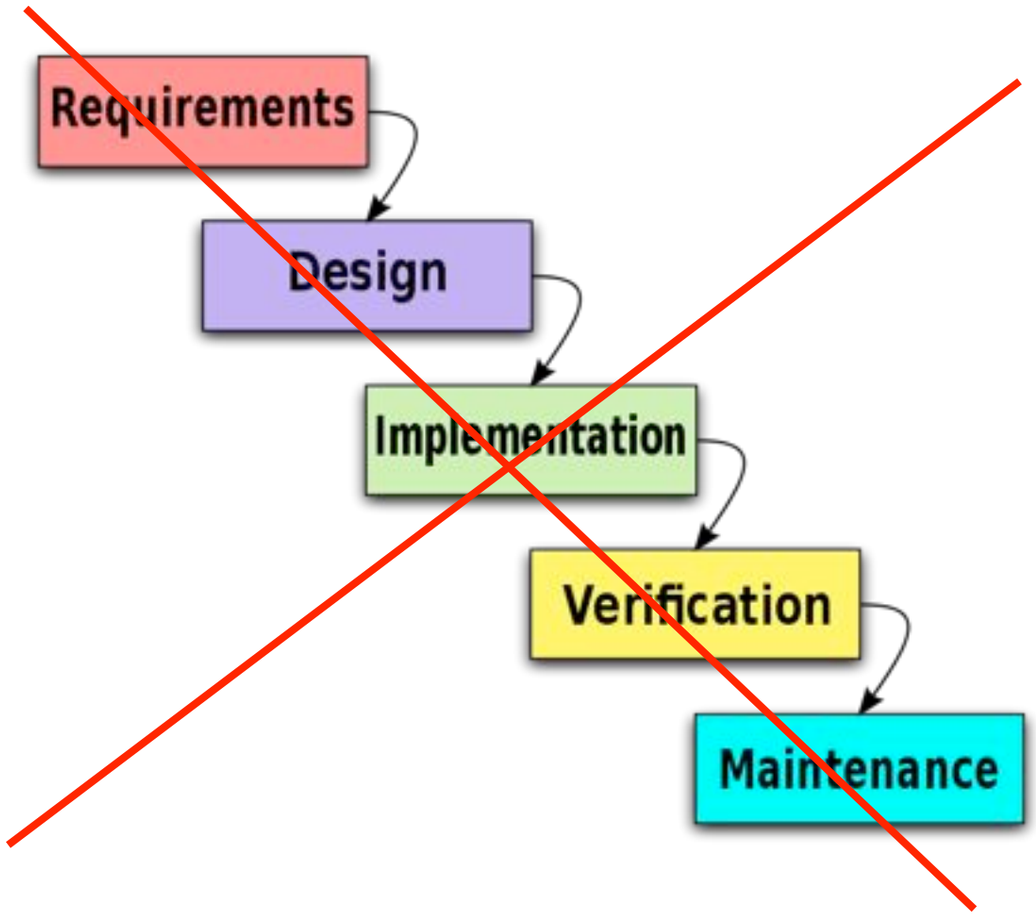
Aim for approximately right rather than accurately wrong

Aim for approximately right rather than accurately wrong

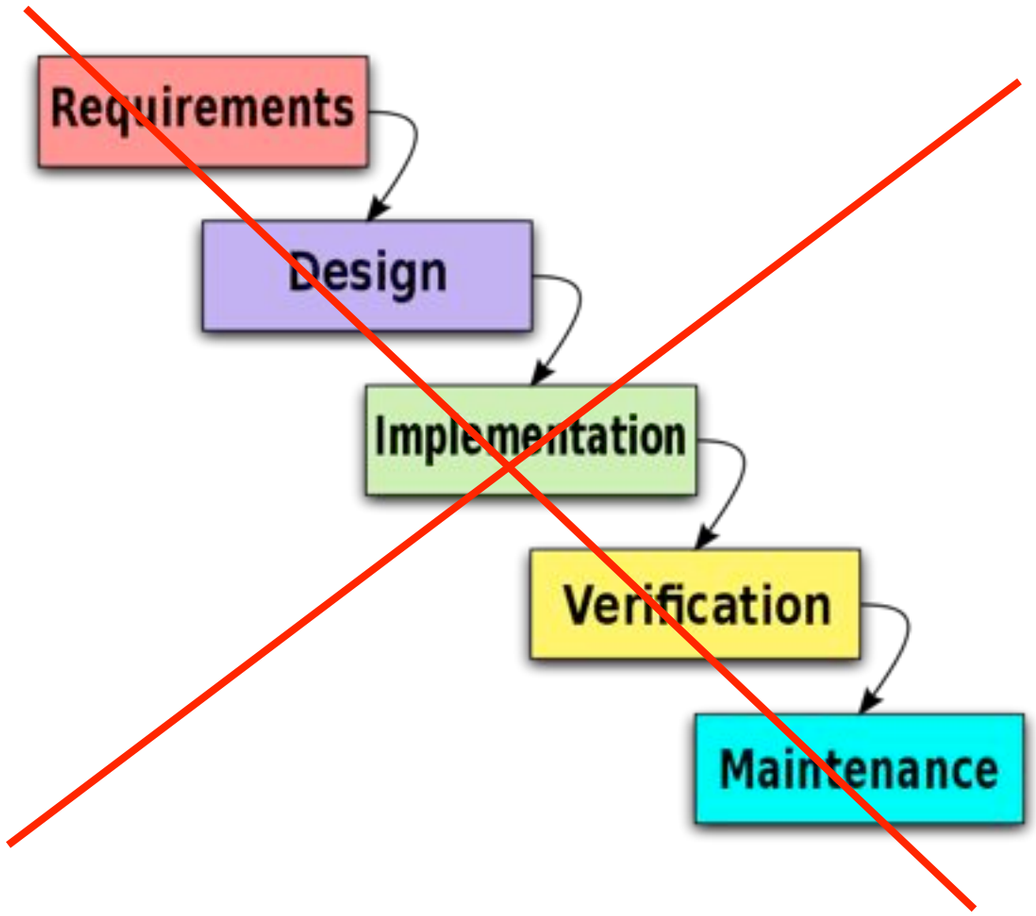
Release early and release often

Follow principles, not processes

Follow principles, not processes

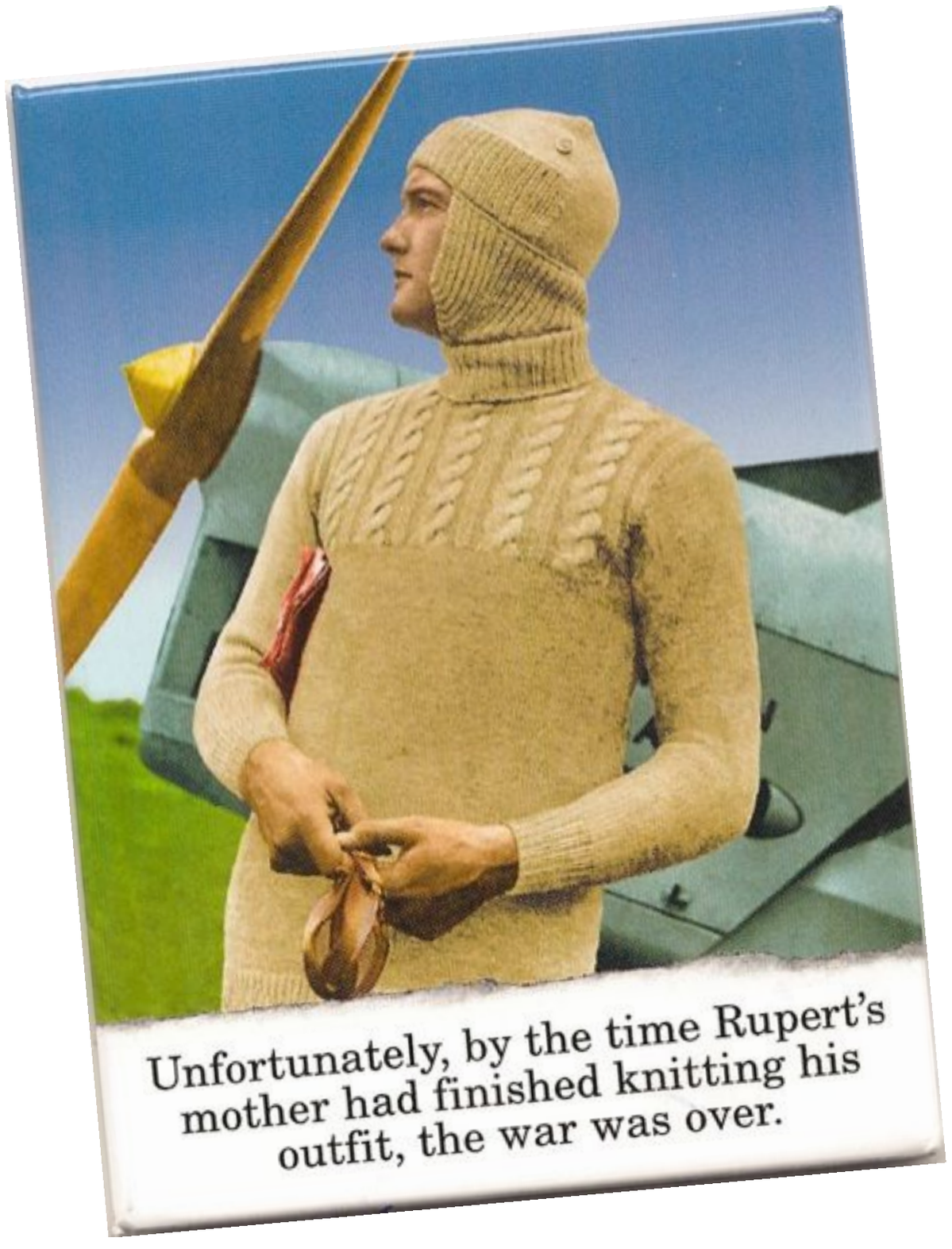


Follow principles, not processes



Timing is everything

Timing is everything



Effective Product Development

- Embrace chaos
- Break the rules
- Respect doers
- Focus on communication
- Introduce slack
- Beware the observer effect
- Constrain innovation
- Reward courage
- Focus on the whole product
- Delay decisions
- Aim for approximately right
- Release early, release often
- Follow principles, not processes
- Timing is everything

About multi-site development

About multi-site development

About multi-site development

- rule 0: avoid multi-site development, if you can...

About multi-site development

- rule 0: avoid multi-site development, if you can...
- time difference is more challenging than geographical separation

About multi-site development

- rule 0: avoid multi-site development, if you can...
- time difference is more challenging than geographical separation
- working across trust boundaries is painful

About multi-site development

- rule 0: avoid multi-site development, if you can...
- time difference is more challenging than geographical separation
- working across trust boundaries is painful
- use continuous integration and automatic testing

About multi-site development

- rule 0: avoid multi-site development, if you can...
- time difference is more challenging than geographical separation
- working across trust boundaries is painful
- use continuous integration and automatic testing
- balance the sites

About multi-site development

- rule 0: avoid multi-site development, if you can...
- time difference is more challenging than geographical separation
- working across trust boundaries is painful
- use continuous integration and automatic testing
- balance the sites
- prefer local decisions

About multi-site development

- rule 0: avoid multi-site development, if you can...
- time difference is more challenging than geographical separation
- working across trust boundaries is painful
- use continuous integration and automatic testing
- balance the sites
- prefer local decisions
- accept duplication of work, beware the share

About multi-site development

- rule 0: avoid multi-site development, if you can...
- time difference is more challenging than geographical separation
- working across trust boundaries is painful
- use continuous integration and automatic testing
- balance the sites
- prefer local decisions
- accept duplication of work, beware the share
- beware of the Cover My Ass game

About multi-site development

- rule 0: avoid multi-site development, if you can...
- time difference is more challenging than geographical separation
- working across trust boundaries is painful
- use continuous integration and automatic testing
- balance the sites
- prefer local decisions
- accept duplication of work, beware the share
- beware of the Cover My Ass game
- avoid contracted interfaces, use mediators

About multi-site development

- rule 0: avoid multi-site development, if you can...
- time difference is more challenging than geographical separation
- working across trust boundaries is painful
- use continuous integration and automatic testing
- balance the sites
- prefer local decisions
- accept duplication of work, beware the share
- beware of the Cover My Ass game
- avoid contracted interfaces, use mediators
- use tools for microcommunication

About multi-site development

- rule 0: avoid multi-site development, if you can...
- time difference is more challenging than geographical separation
- working across trust boundaries is painful
- use continuous integration and automatic testing
- balance the sites
- prefer local decisions
- accept duplication of work, beware the share
- beware of the Cover My Ass game
- avoid contracted interfaces, use mediators
- use tools for microcommunication
- active knowledge management across sites

About multi-site development

- rule 0: avoid multi-site development, if you can...
- time difference is more challenging than geographical separation
- working across trust boundaries is painful
- use continuous integration and automatic testing
- balance the sites
- prefer local decisions
- accept duplication of work, beware the share
- beware of the Cover My Ass game
- avoid contracted interfaces, use mediators
- use tools for microcommunication
- active knowledge management across sites
- focus on system architecture

Summary

Summary

Embrace uncertainty
by responding to change over
following a plan

Summary

Embrace uncertainty
by responding to change over following a plan

Amplify learning by accepting failures and rewarding courage

Summary

Embrace uncertainty
by responding to change over following a plan

Amplify learning by accepting failures and rewarding courage

Become more effective by introducing slack in the system

You can't control what you can't measure.

(Tom DeMarco, 1982)

~~*You can't control what you can't measure.*~~

(Tom DeMarco, 1982)

In my reflective mood, I'm wondering, was its advice correct at the time, is it still relevant, and do I still believe that metrics are a must for any successful software development effort? My answers are no, no, and no

...

Imagine you're trying to control a teenager's upbringing. The very idea of controlling your child ought to make you at least a little bit queasy. Yet the stakes for control couldn't be higher. If you fail in your task, fail utterly, lives can be ruined. So, it's absolutely essential that you not lose your grip entirely.

...

Now apply "You can't control what you can't measure" to the teenager. Most things that really matter—honor, dignity, discipline, personality, grace under pressure, values, ethics, resourcefulness, loyalty, humor, kindness—aren't measurable.

(Tom DeMarco, 2009)

http://www2.computer.org/cms/Computer.org/ComputingNow/homepage/2009/0709/rW_SO_Viewpoints.pdf

Theory X - employees are inherently lazy and will avoid work if they can and that they inherently dislike work.

Theory Y - employees may be ambitious and self-motivated and exercise self-control. It is believed that employees enjoy their mental and physical work duties.

Source: http://en.wikipedia.org/wiki/Theory_X_and_Theory_Y

Seven Lean Principles

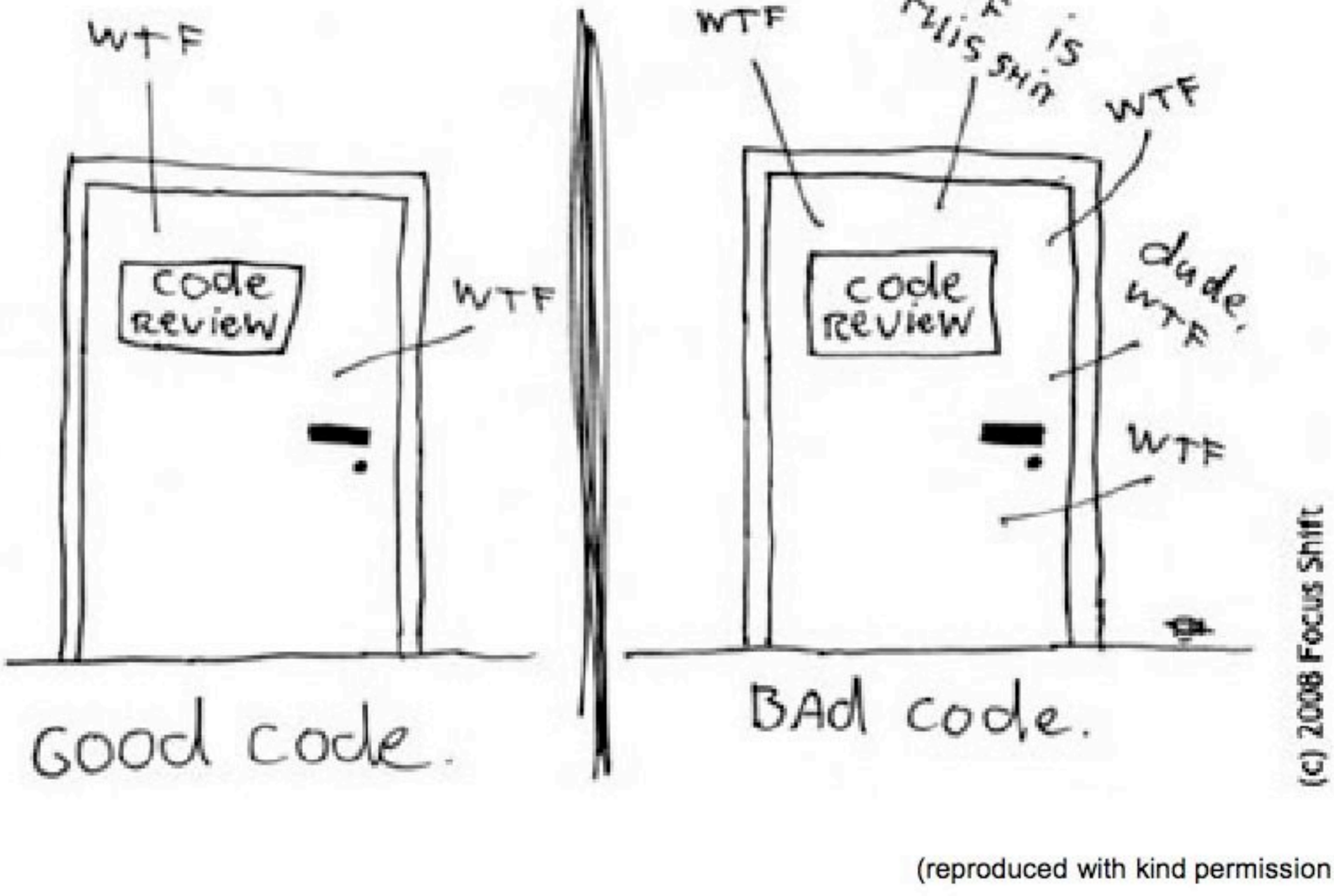
- Eliminate Waste
- Amplify Learning
- Decide as Late as Possible
- Deliver as Fast as Possible
- Empower the Team
- Build Integrity In
- See The Whole

(Poppendieck, 2003)

Control does not always work

Focus on flow

The ONLY VALID MEASUREMENT OF Code QUALITY: WTFs/MINUTE



(reproduced with kind permission of Thom Holwerda)

Make sure that everybody is working towards a common goal.

The more you tighten your grip, Tarkin, the more star systems will slip through your fingers.

(Princess Leia)