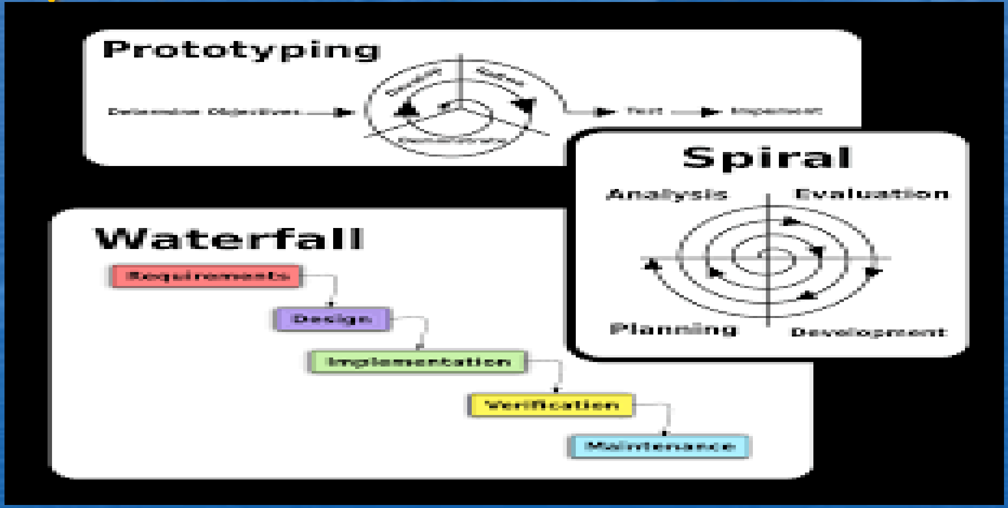
Software Process Models



Software Process Model

- Attempt to organize the software life cycle by
  - ✓ defining activities involved in software production
  - ✓ order of activities and their relationships

- Goals of a software process
  - ✓ standardization, predictability, productivity, high product quality, ability to plan time and budget requirements

Code & Fix

The earliest approach

- Write code
- Fix it to eliminate any errors that have been detected, to enhance existing functionality, or to add new features
- Source of difficulties and deficiencies
  - impossible to predict
  - difficult to manage

Why Models are needed?

- Symptoms of inadequacy: the software crisis
  - scheduled time and cost exceeded
  - user expectations not met
  - poor quality
- The size and economic value of software applications required appropriate "process models"

Goals of Process Model

Determine the order of stages involved in software development and evolution, and to establish the transition criteria for progressing from one stage to the next.

These include completion criteria for the current stage plus choice criteria and entry criteria for the next stage.

Thus a process model addresses the following software project questions:

What shall we do next?
How long shall we continue to do it?

SDLC Model

A framework that describes the activities performed at each stage of a software development project.

Stages of Software Development

Communication

Project initiation

Requirements gathering

Planning

Estimating

Scheduling

Tracking

Construction

Code

Test

Modelling

Analysis

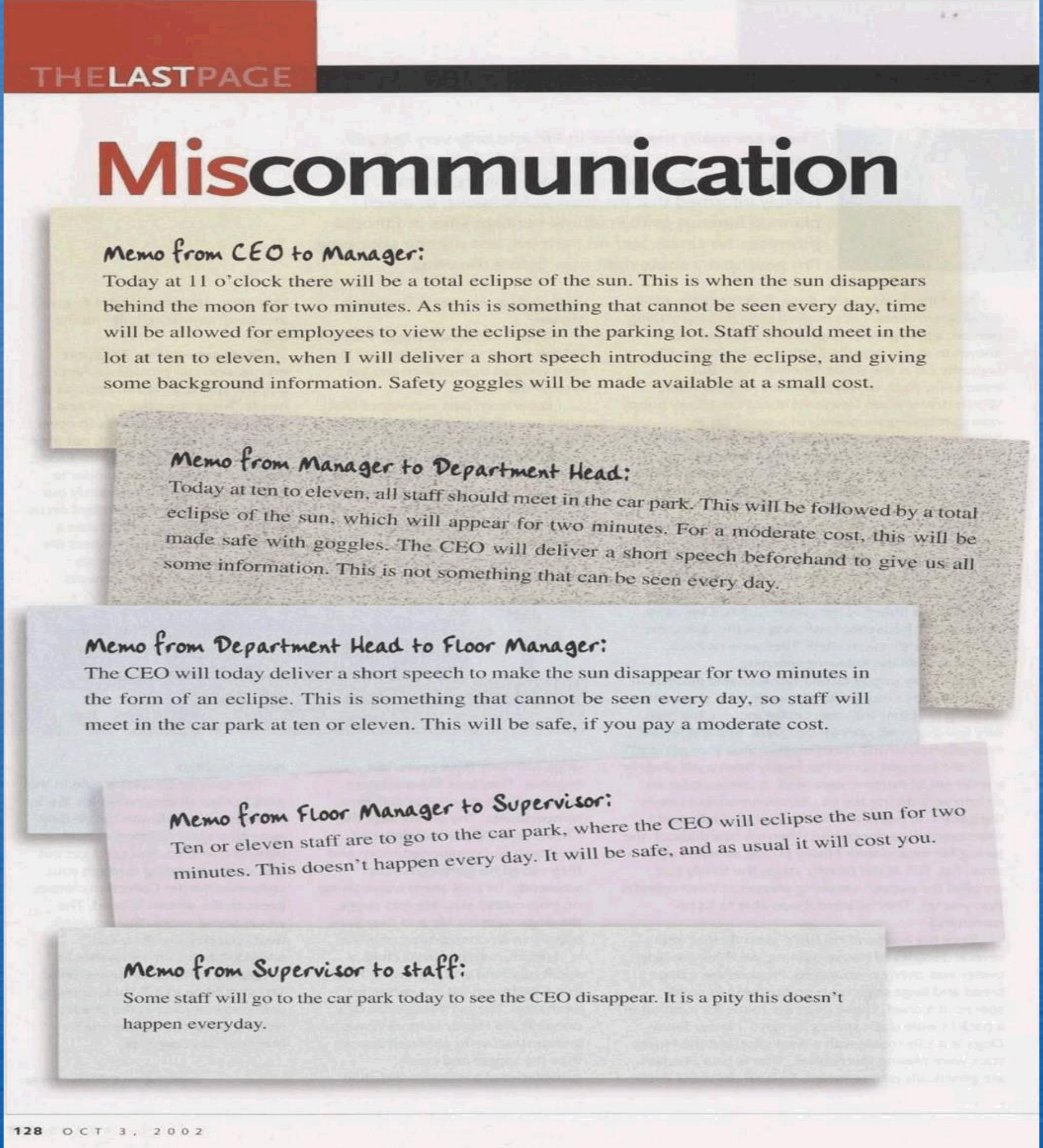
Design

Deployment

Delivery

Support

Feedback



Miscommunication

Memo from CEO to Manager:

Today at 11 o'clock there will be a total eclipse of the sun. This is when the sun disappears behind the moon for two minutes. As this is something that cannot be seen every day, time will be allowed for employees to view the eclipse in the parking lot. Staff should meet in the lot at ten to eleven, when I will deliver a short speech introducing the eclipse, and giving some background information. Safety goggles will be made available at a small cost.

Memo from Manager to Department Head:

Today at ten to eleven, all staff should meet in the car park. This will be followed by a total eclipse of the sun, which will appear for two minutes. For a moderate cost, this will be made safe with goggles. The CEO will deliver a short speech beforehand to give us all some information. This is not something that can be seen every day.

Memo from Department Head to Floor Manager:

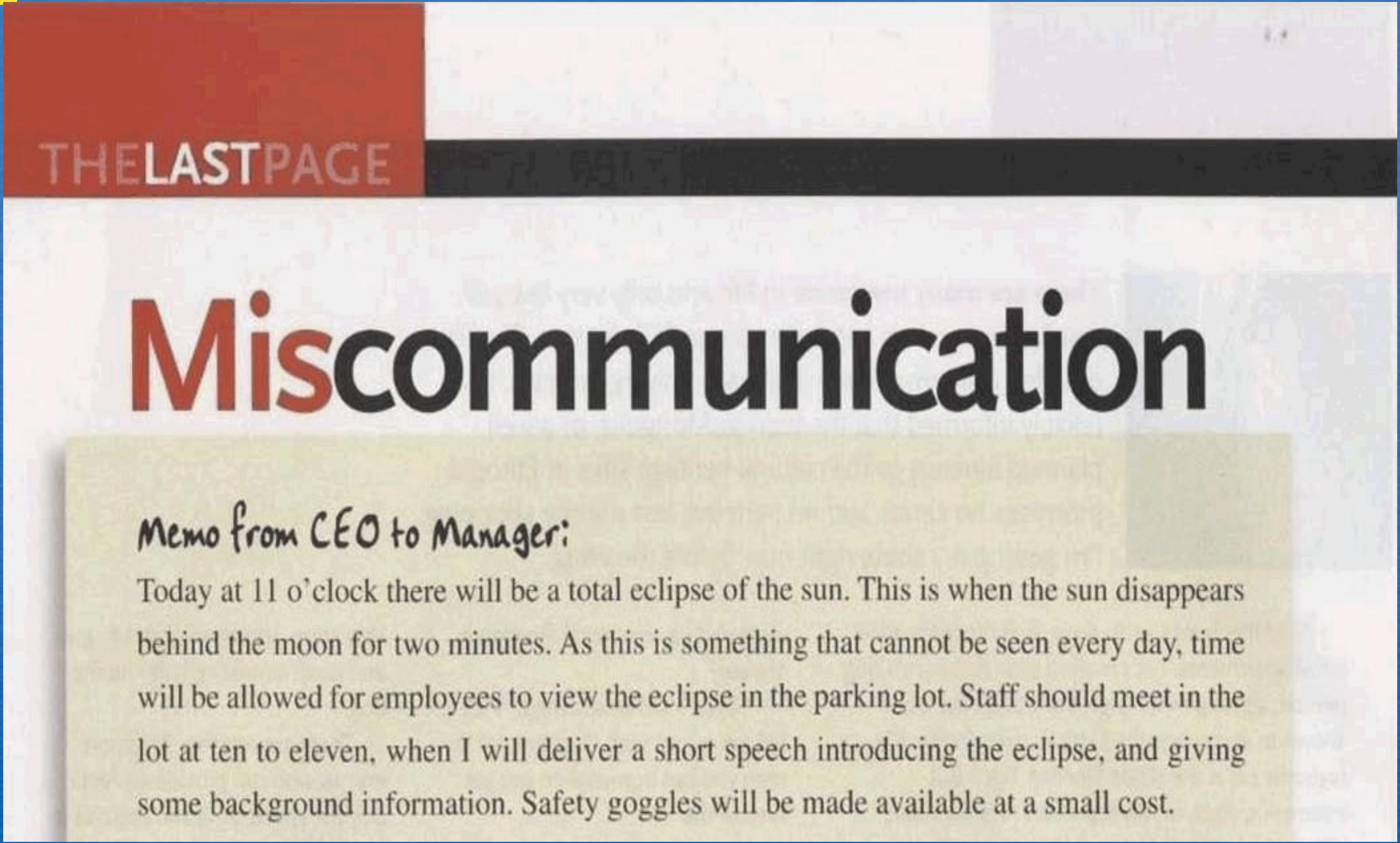
The CEO will today deliver a short speech to make the sun disappear for two minutes in the form of an eclipse. This is something that cannot be seen every day, so staff will meet in the car park at ten or eleven. This will be safe, if you pay a moderate cost.

Memo from Floor Manager to Supervisor:

Ten or eleven staff are to go to the car park, where the CEO will eclipse the sun for two minutes. This doesn't happen every day. It will be safe, and as usual it will cost you.

Memo from Supervisor to staff:

Some staff will go to the car park today to see the CEO disappear. It is a pity this doesn't happen everyday.

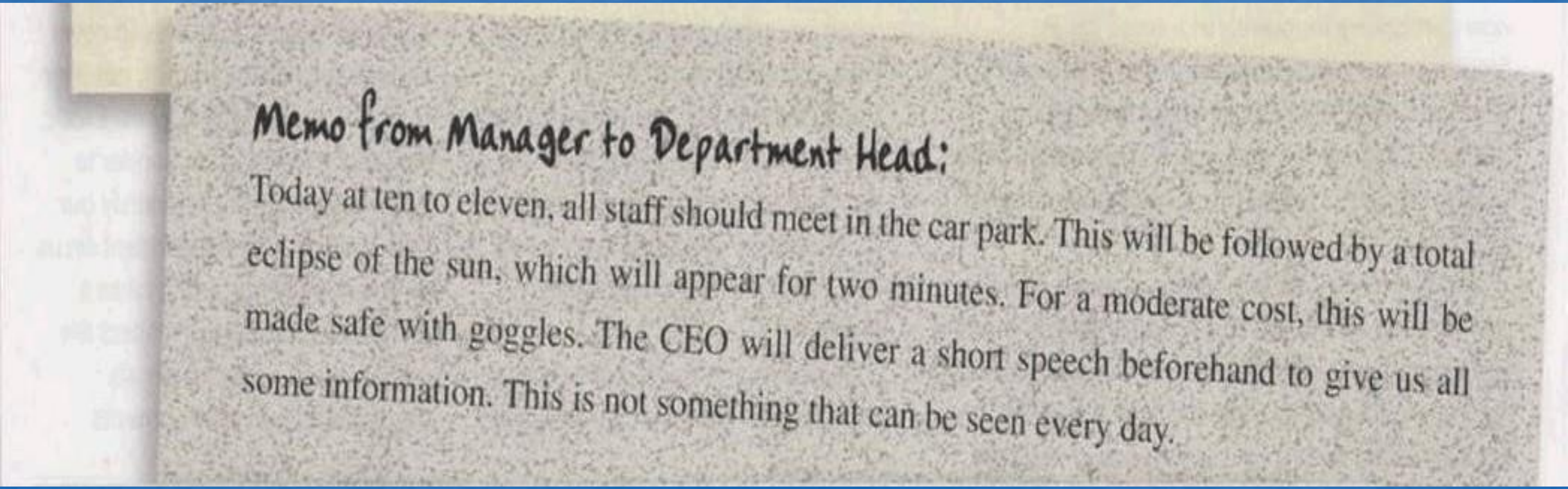


THELASTPAGE

Miscommunication

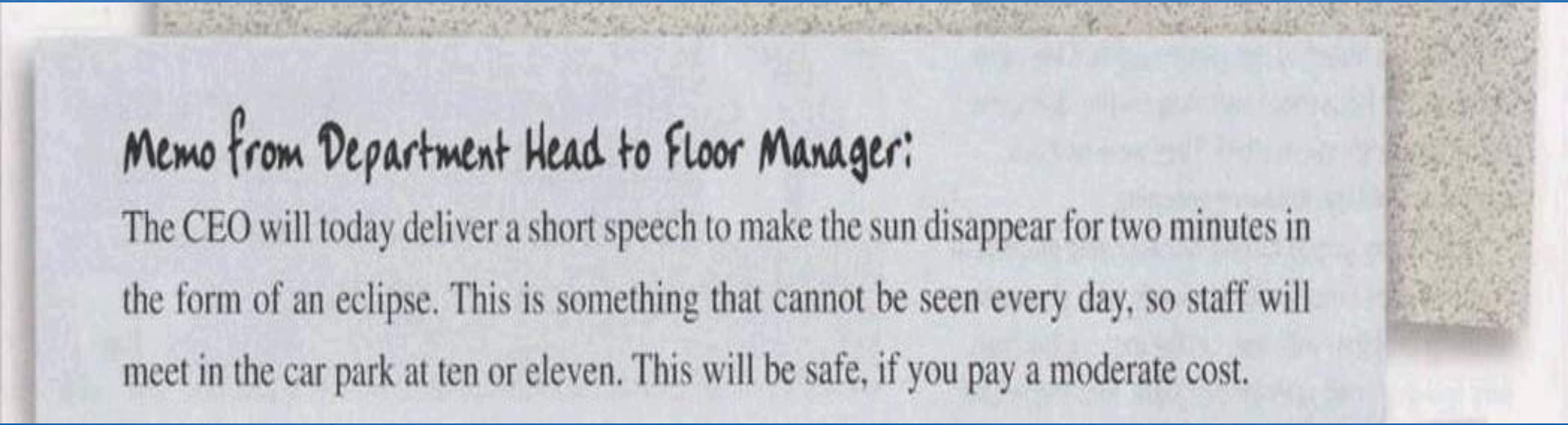
Memo from CEO to Manager:

Today at 11 o'clock there will be a total eclipse of the sun. This is when the sun disappears behind the moon for two minutes. As this is something that cannot be seen every day, time will be allowed for employees to view the eclipse in the parking lot. Staff should meet in the lot at ten to eleven, when I will deliver a short speech introducing the eclipse, and giving some background information. Safety goggles will be made available at a small cost.



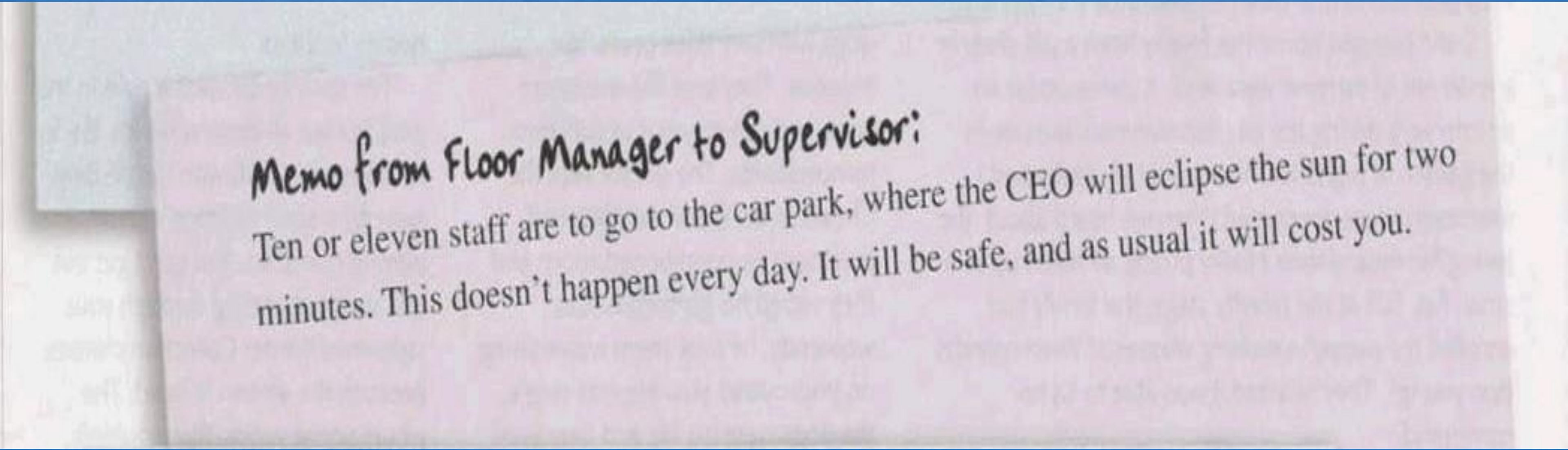
Memo from Manager to Department Head:

Today at ten to eleven, all staff should meet in the car park. This will be followed by a total eclipse of the sun, which will appear for two minutes. For a moderate cost, this will be made safe with goggles. The CEO will deliver a short speech beforehand to give us all some information. This is not something that can be seen every day.



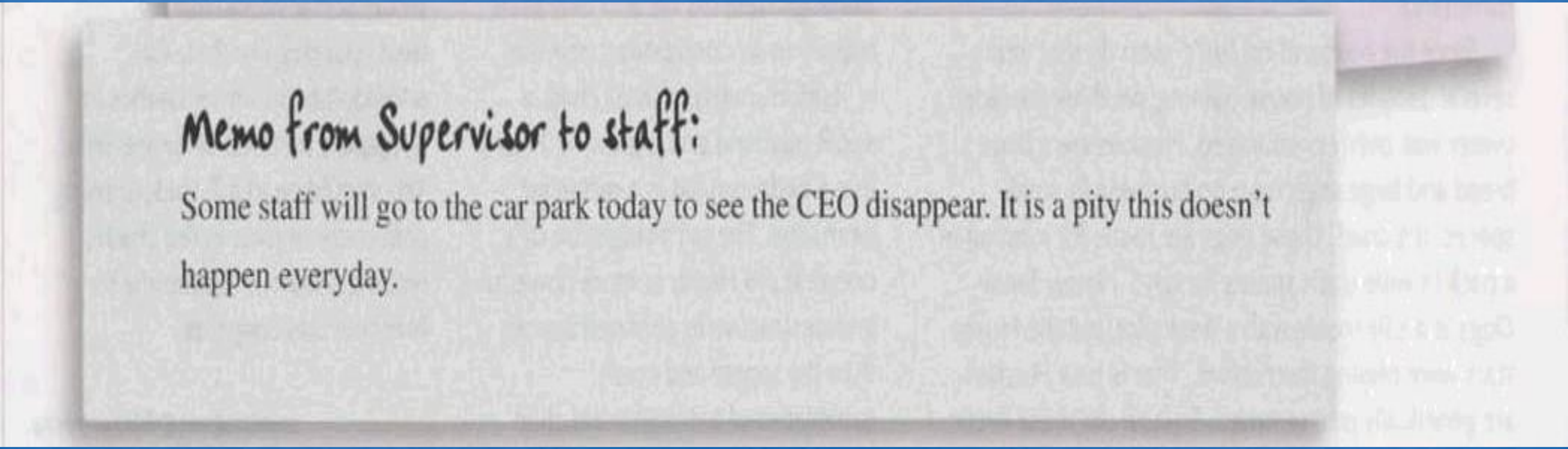
Memo from Department Head to Floor Manager:

The CEO will today deliver a short speech to make the sun disappear for two minutes in the form of an eclipse. This is something that cannot be seen every day, so staff will meet in the car park at ten or eleven. This will be safe, if you pay a moderate cost.



Memo from Floor Manager to Supervisor:

Ten or eleven staff are to go to the car park, where the CEO will eclipse the sun for two minutes. This doesn't happen every day. It will be safe, and as usual it will cost you.

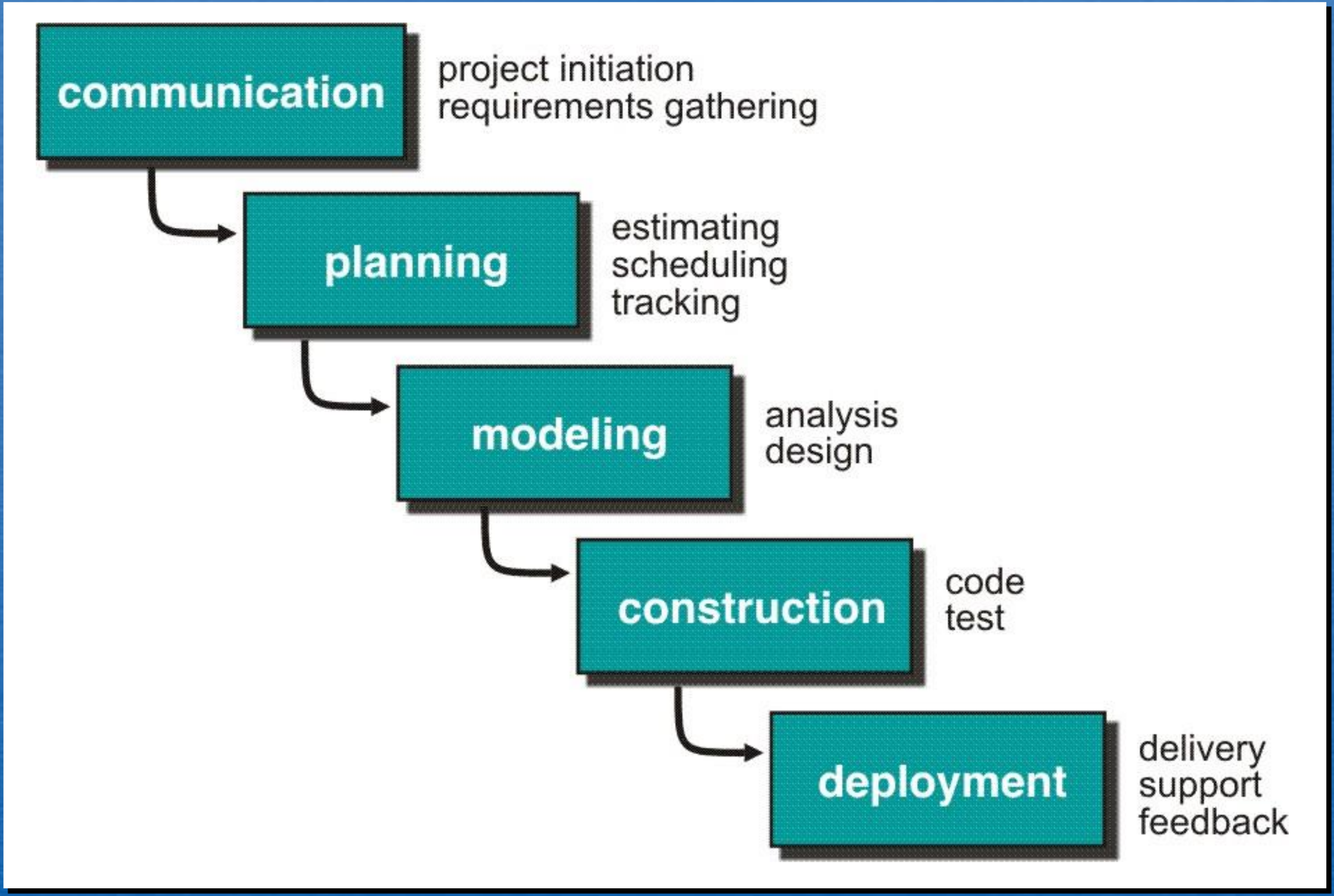


Memo from Supervisor to staff:

Some staff will go to the car park today to see the CEO disappear. It is a pity this doesn't happen everyday.

Just for Laugh

The Waterfall Model



communication

project initiation
requirements gathering

planning

estimating
scheduling
tracking

modeling

analysis
design

construction

code
test

deployment

delivery
support
feedback

Waterfall Strengths

- Easy to understand, easy to use
- Milestones are well understood
- Sets requirements stability
- Good for management control (plan, staff, track)
- Works well when quality is more important than cost or schedule

Waterfall Deficiencies

- All requirements must be known upfront
- Deliverables created for each phase are considered frozen – inhibits flexibility
- Does not reflect problem-solving nature of software development – iterations of phases
- Integration is one big bang at the end
- Little opportunity for customer to preview the system (it may be too late)

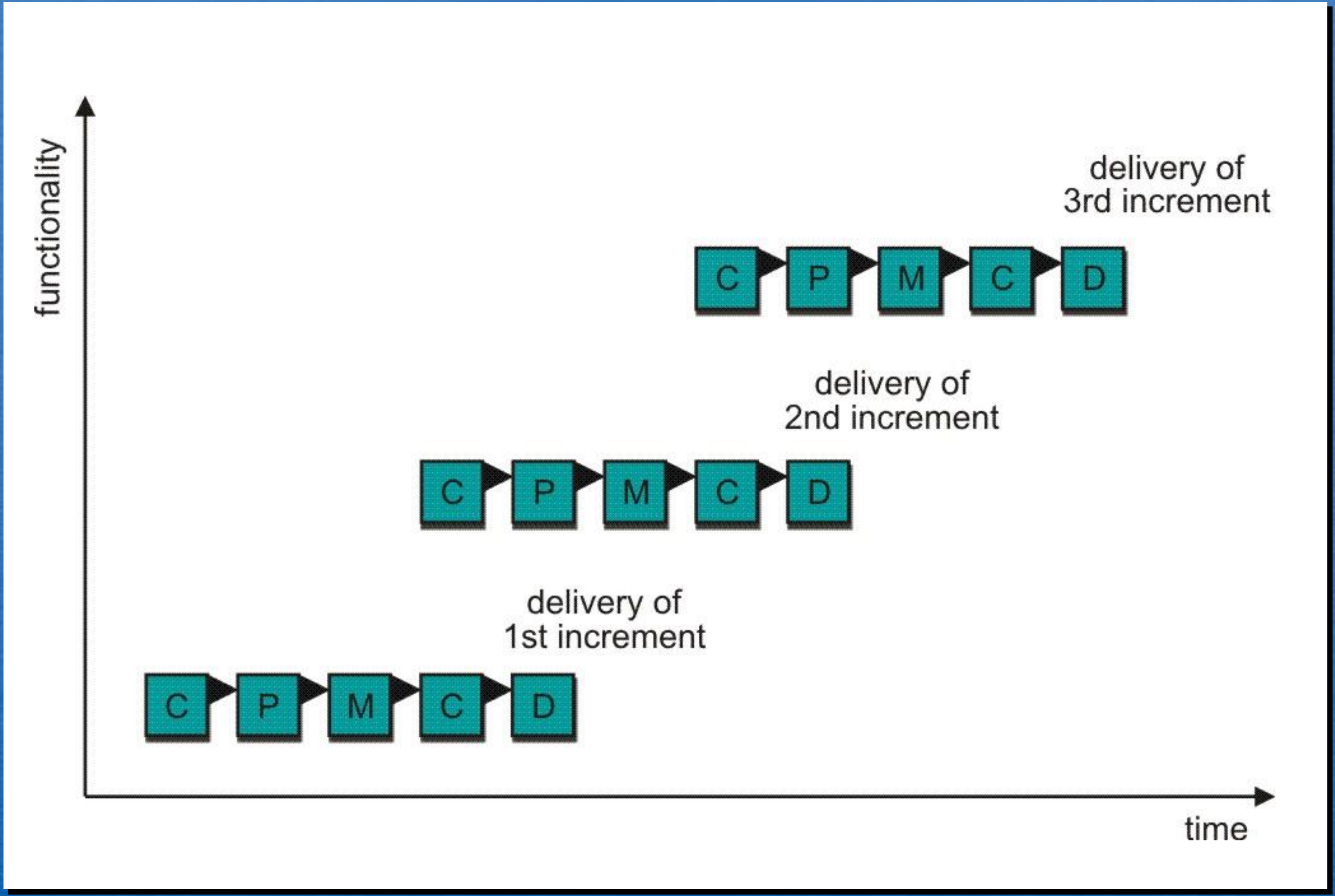
When to use the Waterfall Model?

- Requirements are very well known
- Product definition is stable
- Technology is understood
- New version of an existing product
- Porting an existing product to a new platform.

  - High risk for new systems because of specification and design problems.

  - Low risk for well-understood developments using familiar technology.

Incremental Model



Incremental Model Strengths

- Develop high-risk or major functions first

- Each increment delivers an operational product
- Customer can respond to each increment

- Lowers initial delivery cost
- Initial product delivery is faster

- Customers get important functionality early
- Risk of changing requirements is reduced

Incremental Model Weaknesses

- Requires good planning and design
- Requires early definition of a complete and fully functional system to allow for the definition of increments
- Well-defined module interfaces are required (some will be developed long before others)
- Total cost of the complete system is not lower

When to use the Incremental Model?

- Risk, funding, schedule, program complexity, or need for early realization of benefits.

- Most of the requirements are known up-front but are expected to evolve over time

- A need to get basic functionality to the market early

- On projects which have lengthy development schedules

- On a project with new technology

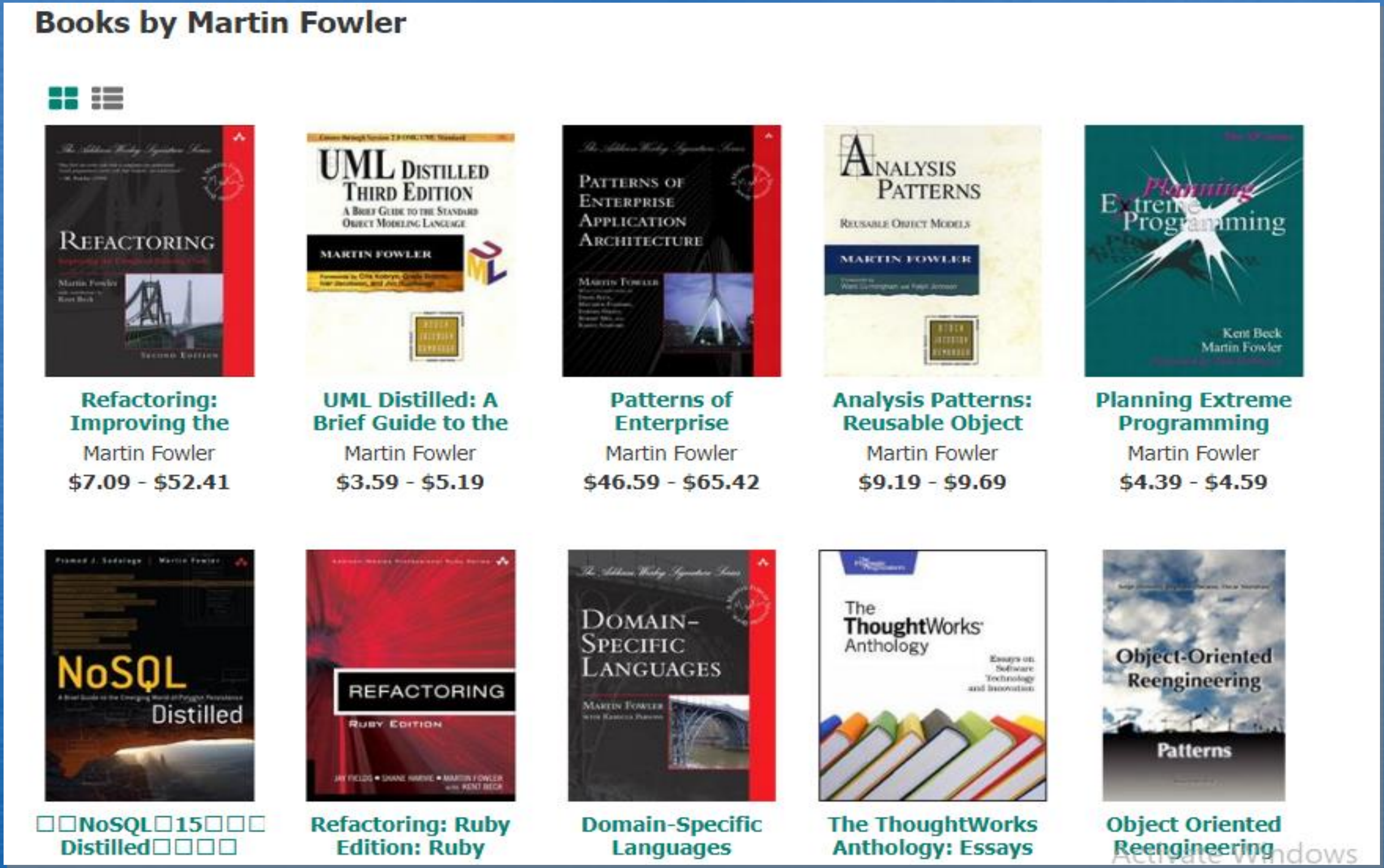
“

When to use iterative development?

You should use iterative development only on projects that you want to succeed.

Martin Fowler

Author and programmer


Refactoring
Martin Fowler
UML Distilled Third Edition
A Brief Guide to the Standard Object Modeling Language
Martin Fowler
Patterns of Enterprise Application Architecture
Martin Fowler
Analysis Patterns
Reusable Object Models
Martin Fowler
Planning Extreme Programming
Kent Beck
Martin Fowler
NoSQL
Distilled
Refactoring
Ruby Edition
Domain-Specific Languages
Martin Fowler
The ThoughtWorks Anthology
Essays on Software Technology and Innovation
Object-Oriented Reengineering
Patterns

Books by Martin Fowler

Refactoring: Improving the
Martin Fowler
$7.09 - $52.41

UML Distilled: A Brief Guide to the
Martin Fowler
$3.59 - $5.19

Patterns of Enterprise
Martin Fowler
$46.59 - $65.42

Analysis Patterns: Reusable Object
Martin Fowler
$9.19 - $9.69

Planning Extreme Programming
Martin Fowler
$4.39 - $4.59

□□NoSQL□15□□□ Distilled□□□□

Refactoring: Ruby Edition: Ruby

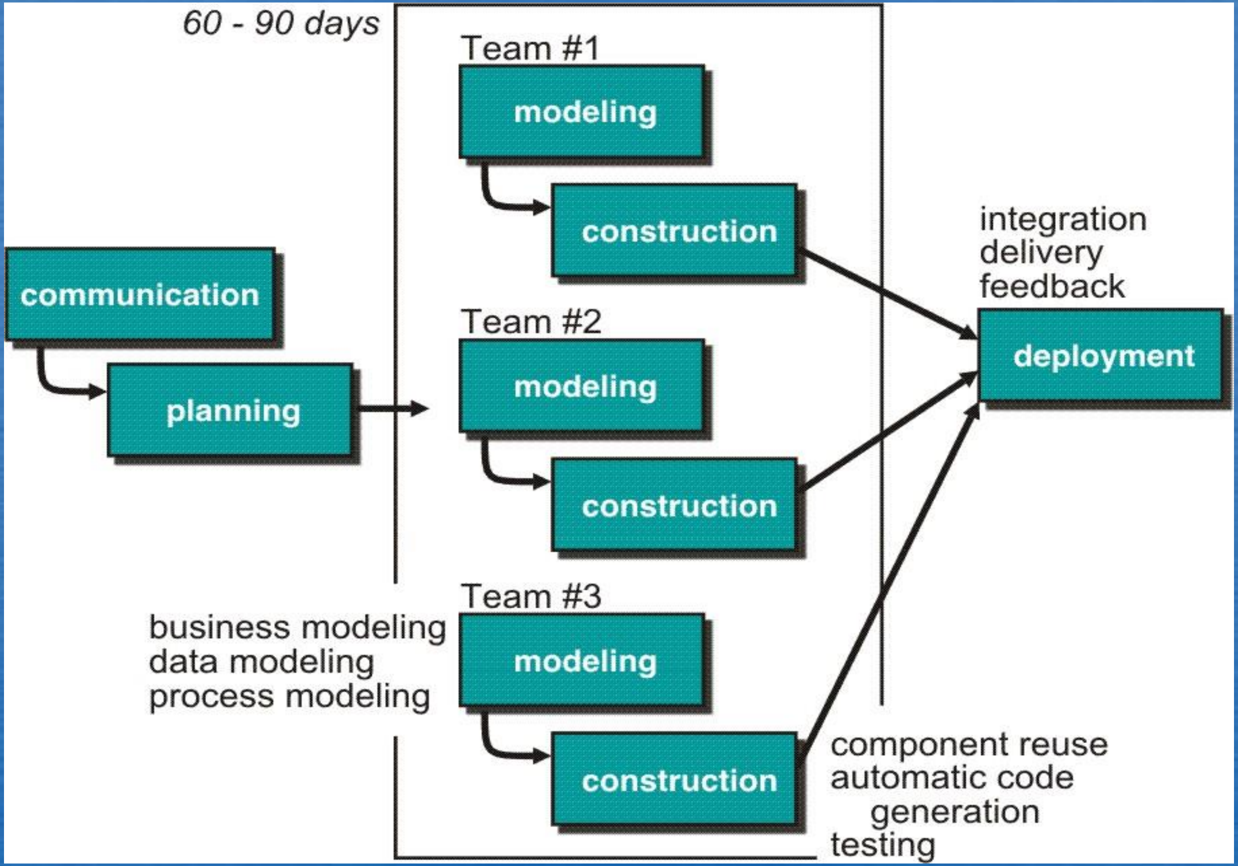
Domain-Specific Languages

The ThoughtWorks Anthology: Essays

Object Oriented Reengineering

Activate Windows

RAD Model



Rapid Application Development – RAD

RAD Strengths

- Reduced cycle time and improved productivity with fewer people means lower costs

- Time-box approach mitigates cost and schedule risk

- Customer involved throughout the complete cycle minimizes risk of not achieving customer satisfaction and business needs

RAD Weaknesses

- Accelerated development process must give quick responses to the user

- Risk of never achieving closure

- Hard to use with legacy systems

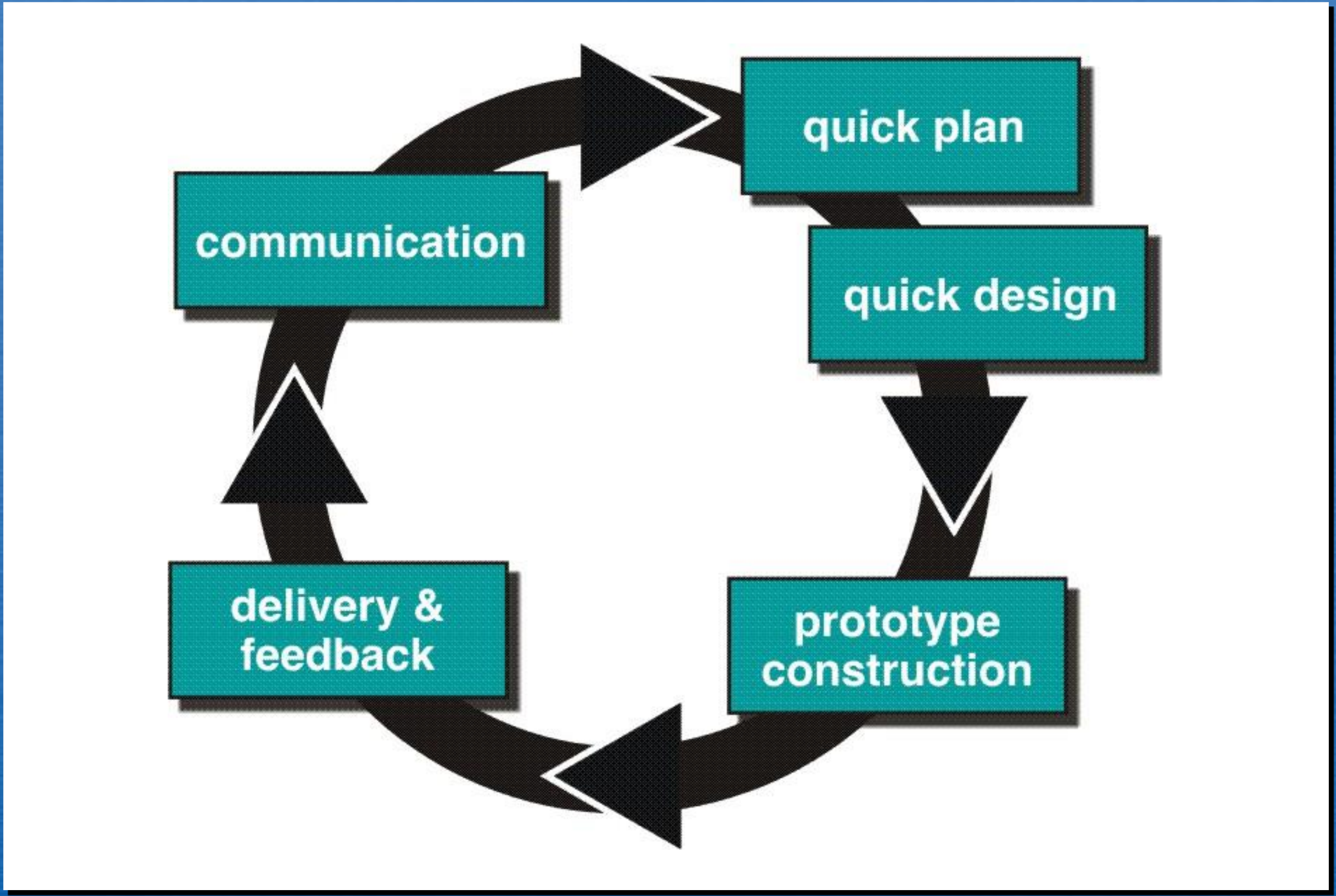
- Requires a system that can be modularized

- Developers and customers must be committed to rapid-fire activities in an abbreviated time frame.

When to use RAD?

- Reasonably well-known requirements
- User involved throughout the life cycle
- Project can be time-boxed
- Functionality delivered in increments
- Low technical risks
- System can be modularized

Evolutionary Models: Prototyping


quick plan
quick design
prototype construction
delivery & feedback
communication