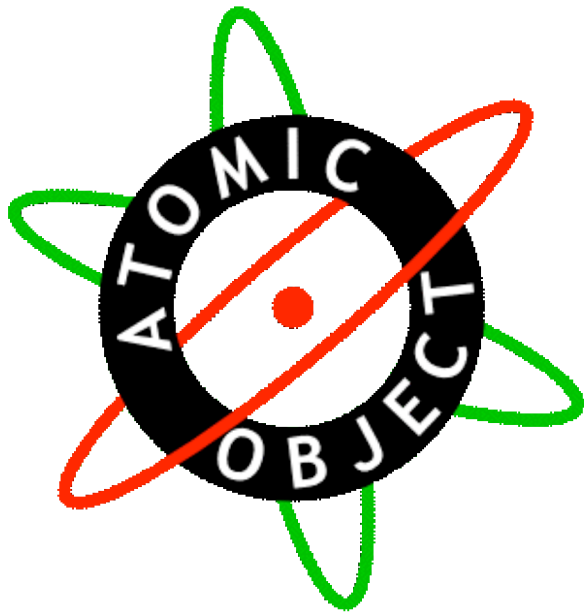
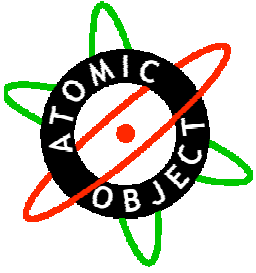


# Sane software development



Carl Erickson, PhD  
Atomic Object LLC



# State of our industry

## 2004 Standish Group study

- 30% total failure, cancelled

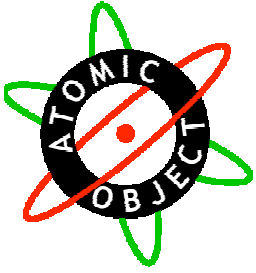
- 50% over budget

- 90% late

## Chaos report, 1994

- 31% cancelled

- 53% more than 2x over budget

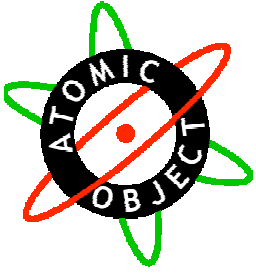


# Status quo

"Our challenge is to get our software to the point that people expect it to work instead of expecting it to fail."

Jim Larus, leader of software quality project at Microsoft Research

MIT Technology Review, April 2003



# The price we pay

NIST - industry losses

**\$60 billion** a year

Doesn't even count...

government

education

home

loss of life

**Therac-25**

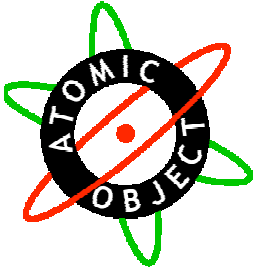
**Denver Airport**

**Ariane 5**

**USS Yorktown**

**FBI Virtual Case File**





# Successful software projects

Paid for itself per the business needs (ROI)

Met important deadlines

Produced software that

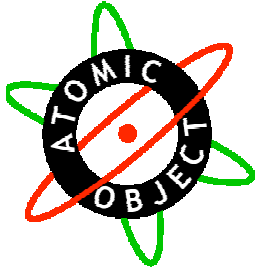
- satisfies end users

- is stable and reliable

- performs acceptably

- is extensible and maintainable

Left developers and managers happy



# To the rescue?

HLL

Structured D/P

4GL

AI

OO

Client/Server

Java

N Tier

**Visual XXX**

**.NET**

AOP

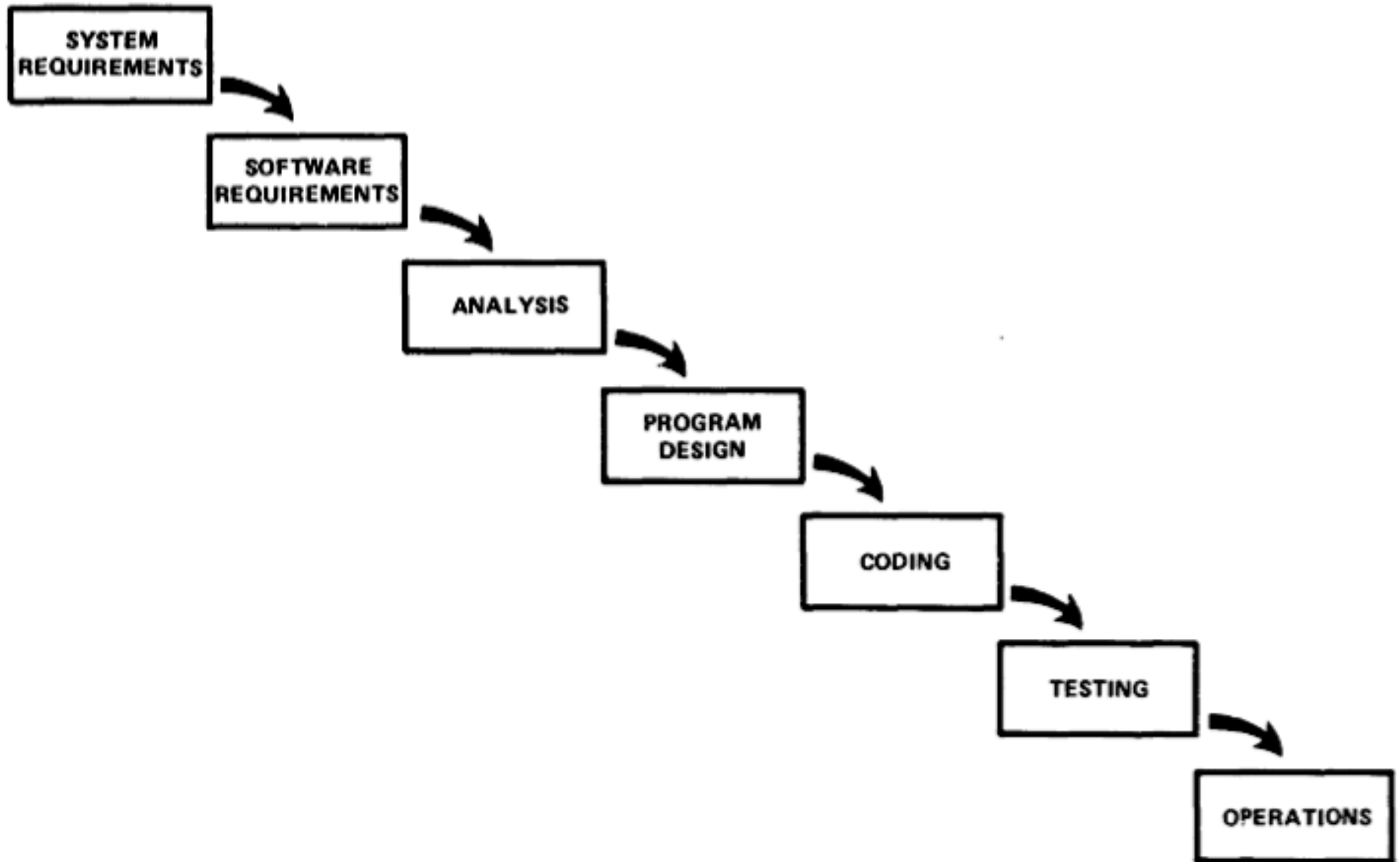
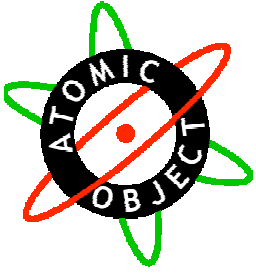


Figure 2. Implementation steps to develop a large computer program for delivery to a customer.



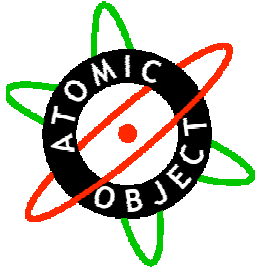
## Page 2

"I believe in this concept, but the implementation described above is risky and invites failure."

Winston Royce

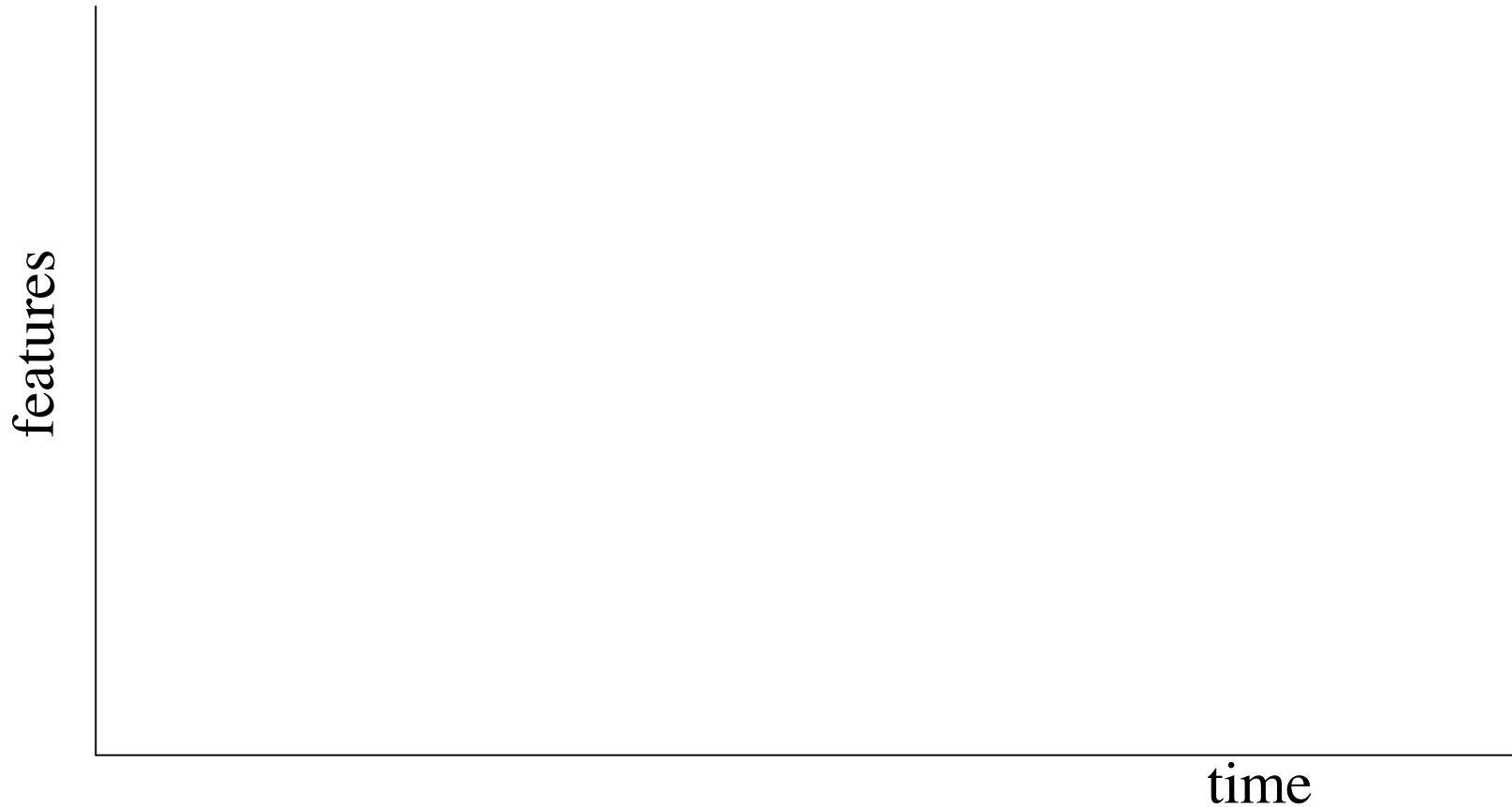
"Managing the Development of Large Software Systems"

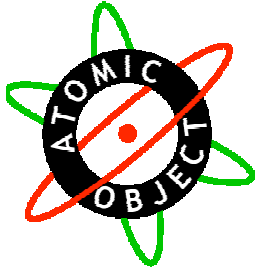
IEEE WESCON, August 1970



(Ron Jeffries inspired these graphs)

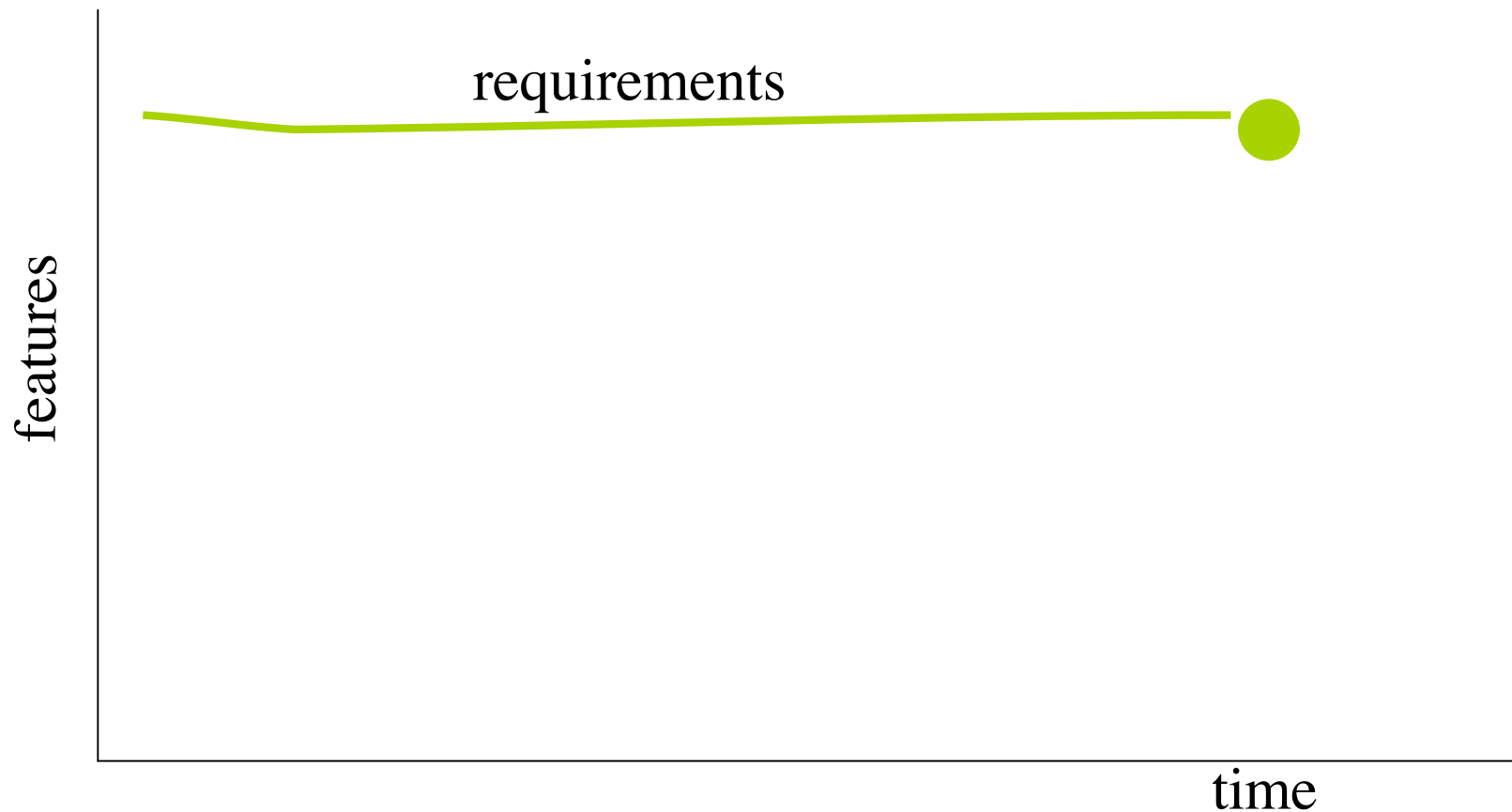
# Waterfall according to plan

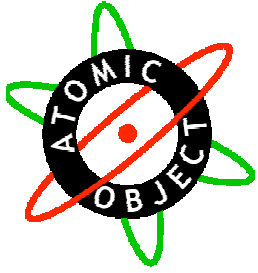




(Ron Jeffries inspired these graphs)

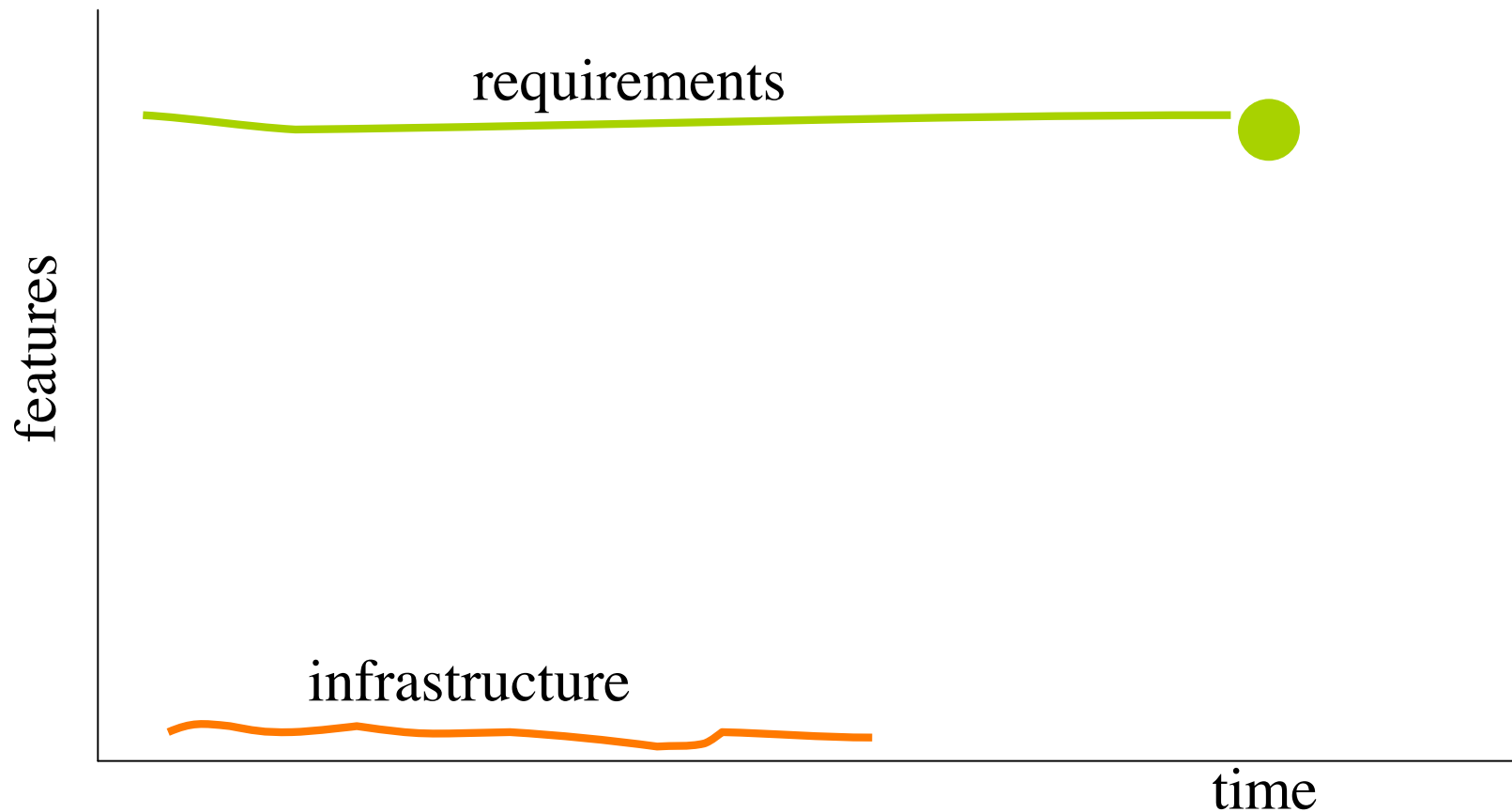
# Waterfall according to plan

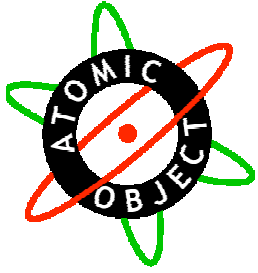




(Ron Jeffries inspired these graphs)

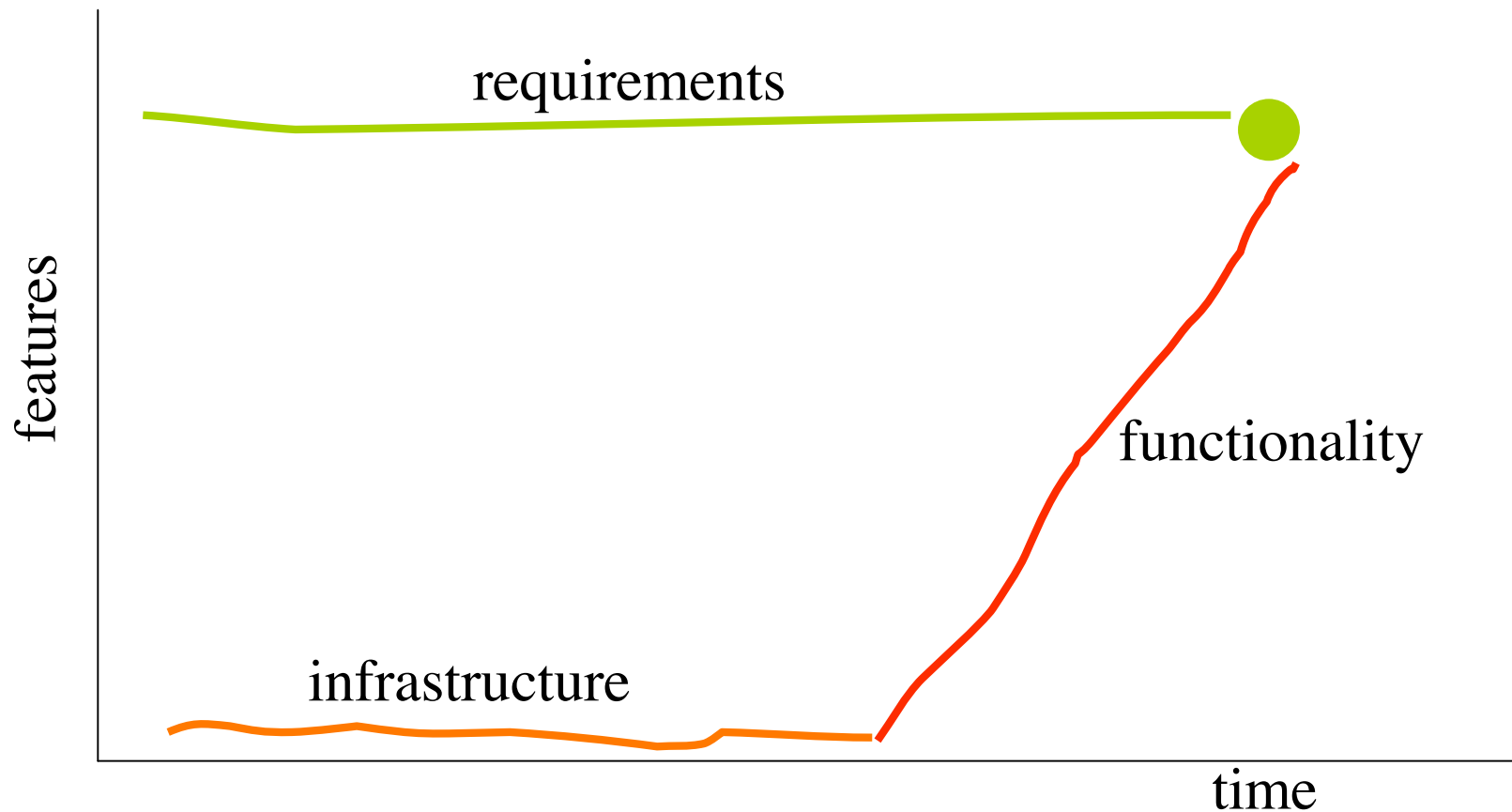
# Waterfall according to plan



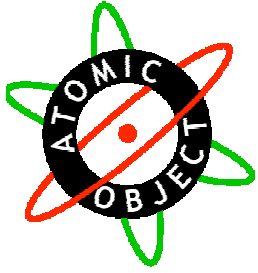


(Ron Jeffries inspired these graphs)

# Waterfall according to plan

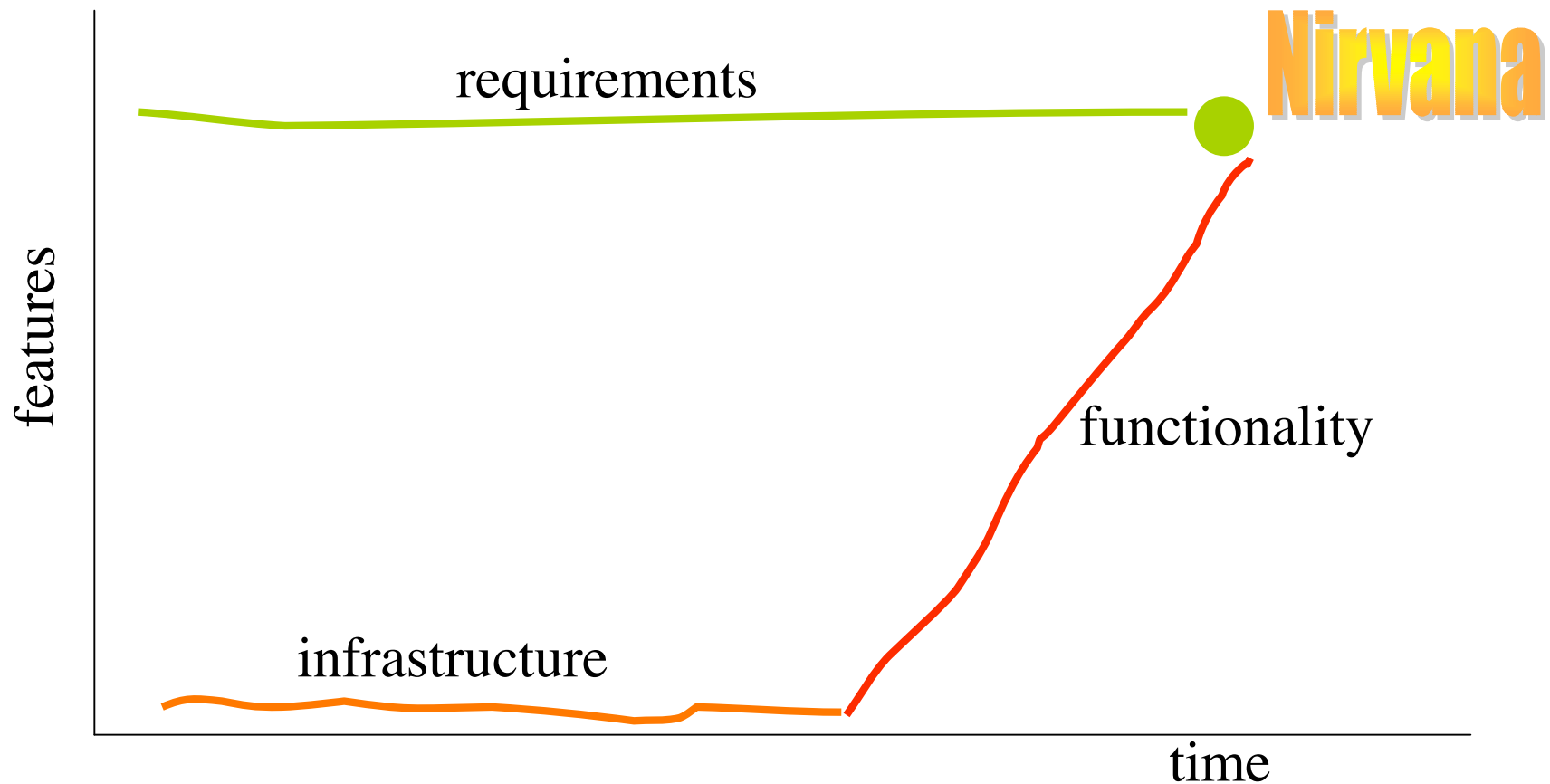


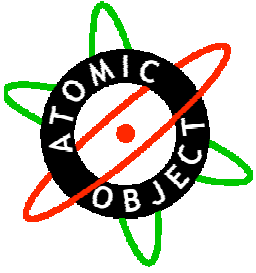




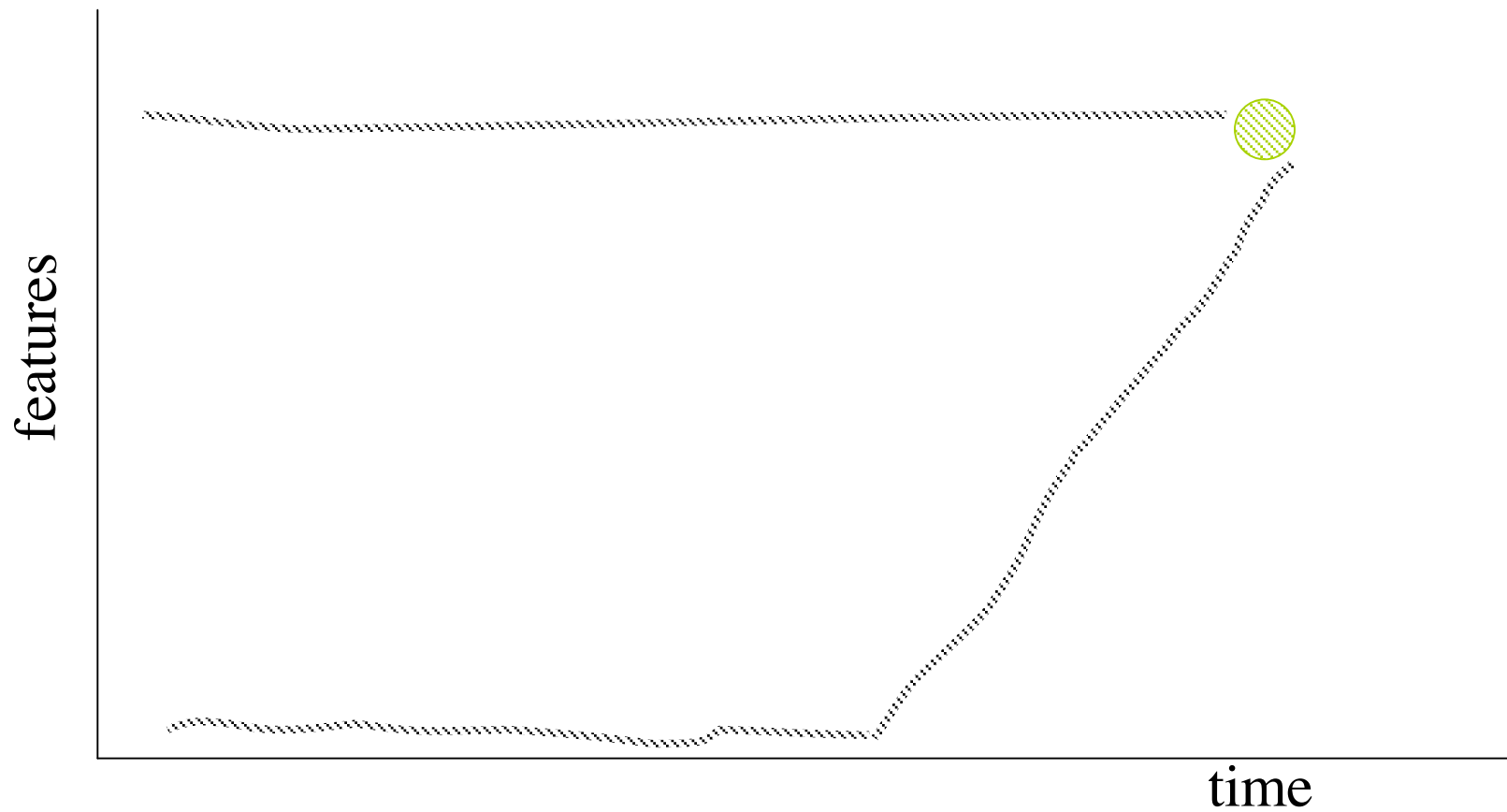
(Ron Jeffries inspired these graphs)

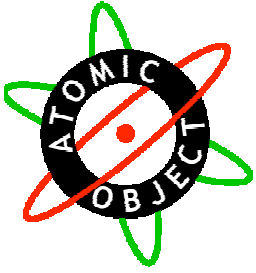
# Waterfall according to plan



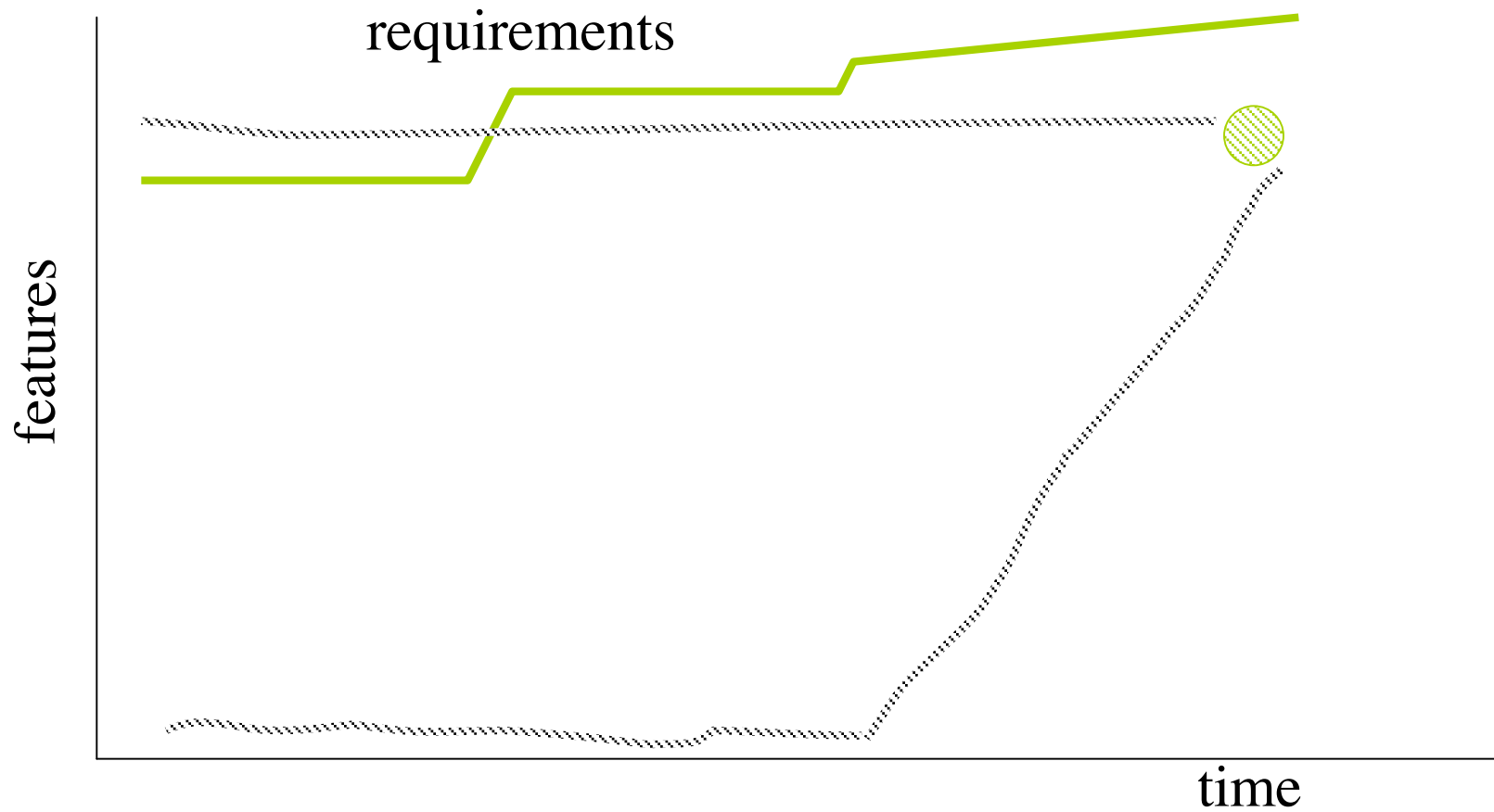


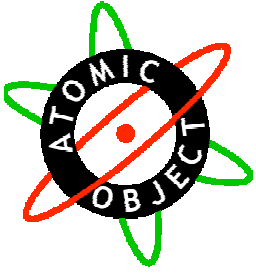
# Waterfall reality



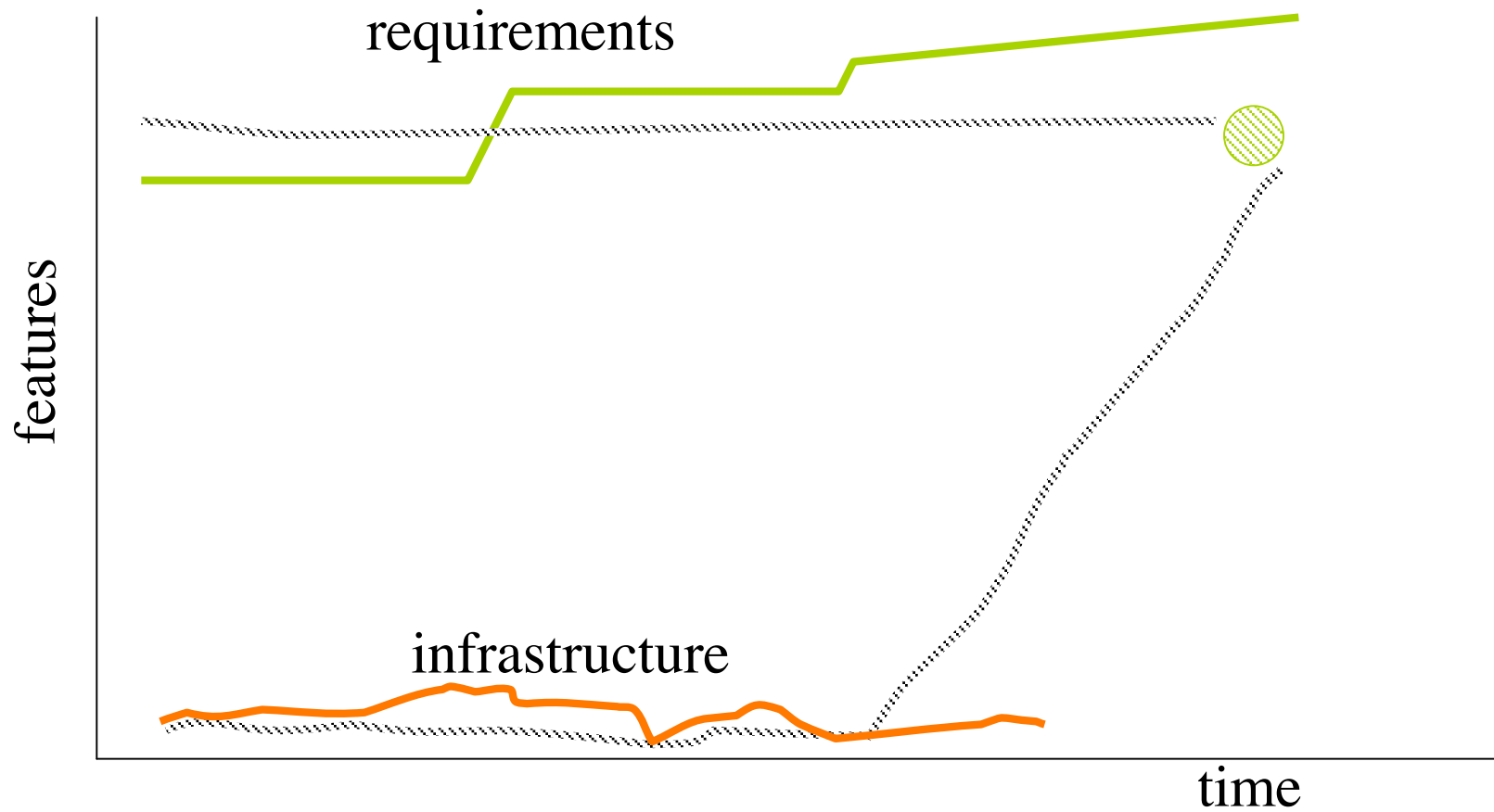


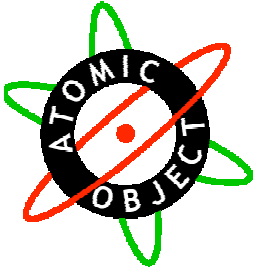
# Waterfall reality



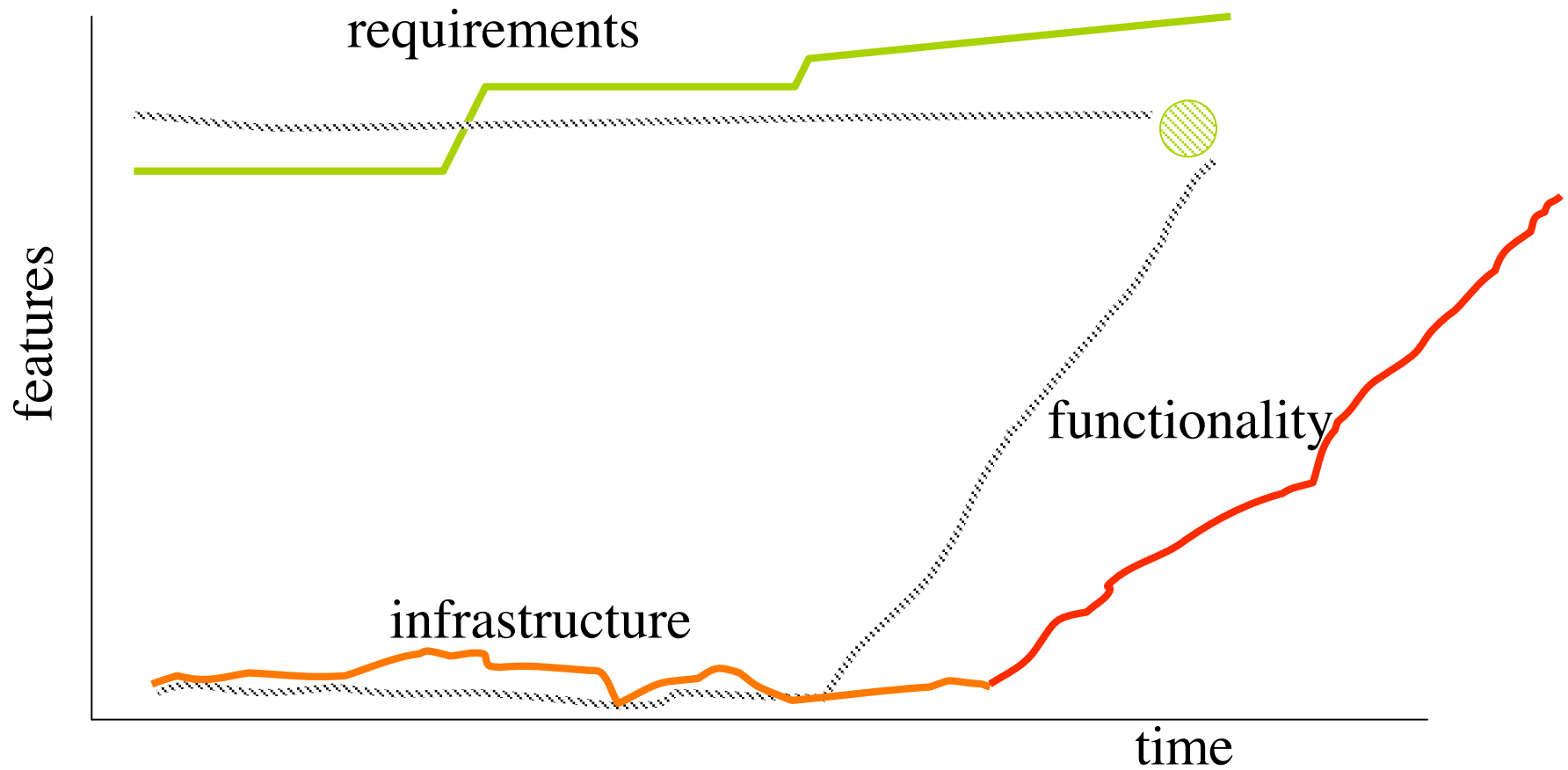


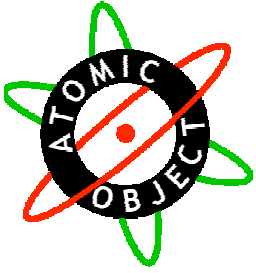
# Waterfall reality



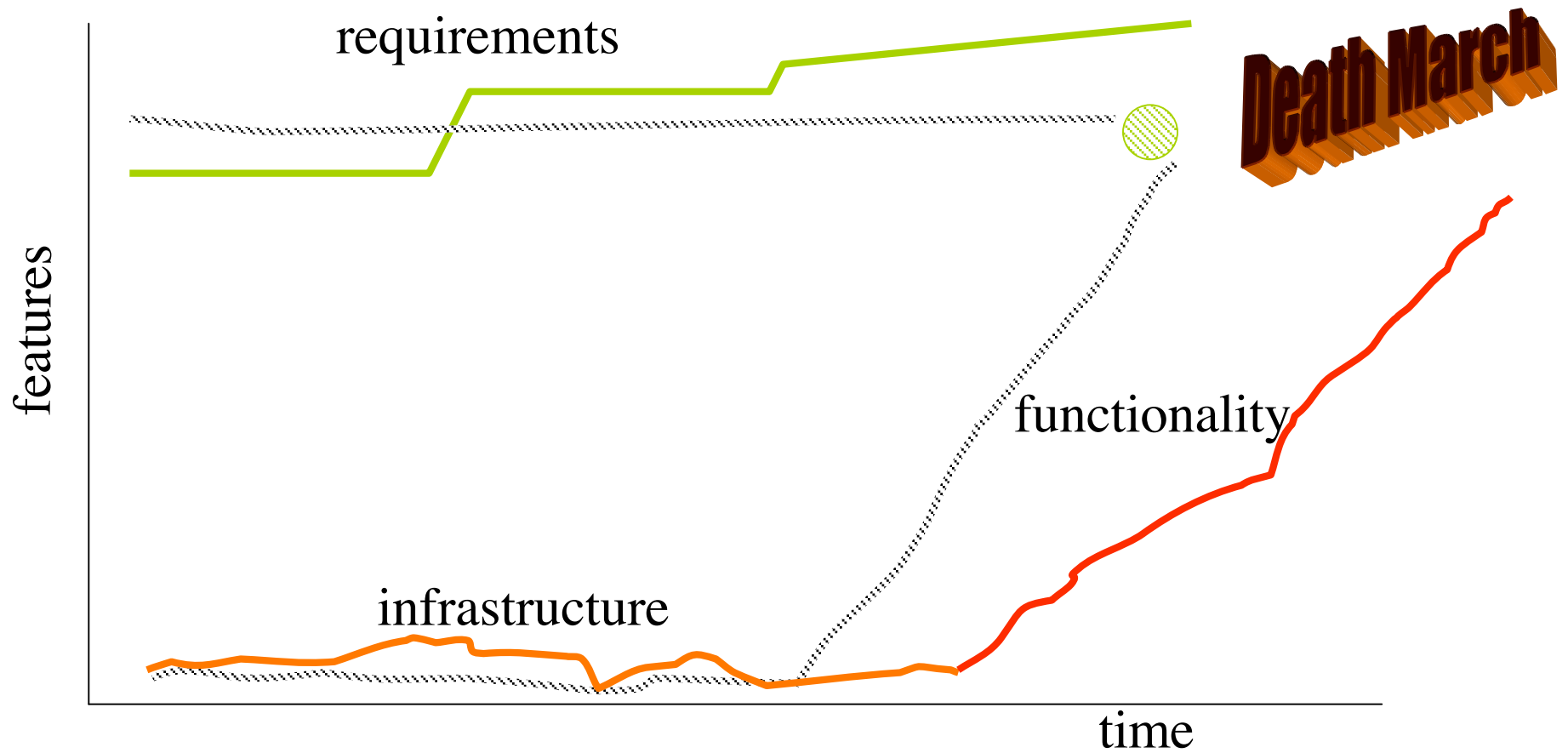


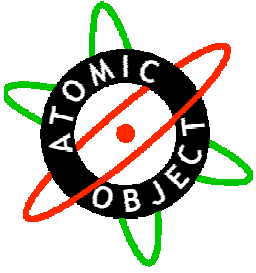
# Waterfall reality





# Waterfall reality

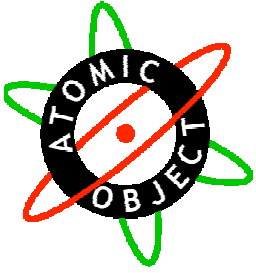




# Agile Alternative

- Learn enough to start
- Express problem in features
- Maintain working system, fully tested, ready to ship
- Deliver features consistently
- Evolve the design as you learn

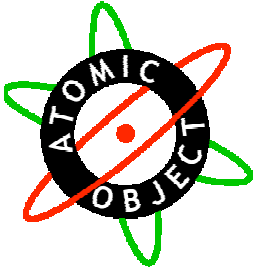




# Extreme Programming?

2am, marathon, Cheetos,  
hacking, greasy,  
undocumented, brilliant,  
tests?, mysterious,  
Jolt Cola, protective,  
ad-hoc, brittle, cubicle,  
fear change, cowboy,  
overly elegant, code ego,  
real soon now, stress  
difficult to talk to





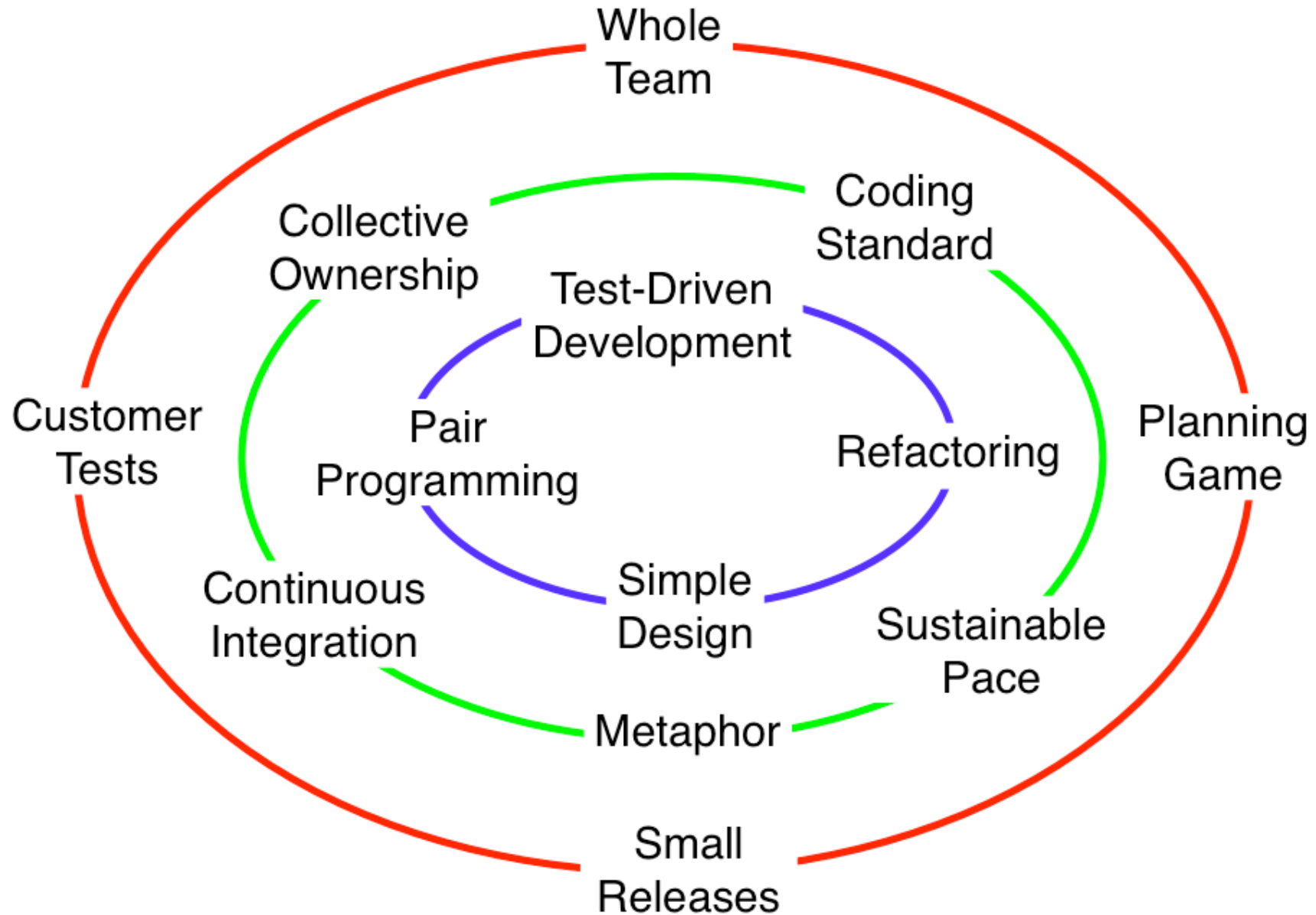
# Extreme Programming!

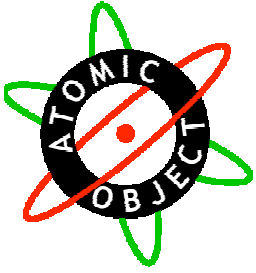
2am, marathon, Cheetos,  
hacking, greasy,  
undocumented, brilliant,  
tests?, mysterious,  
Jolt Cola, protective,  
ad-hoc, brittle, cubicle,  
fear change, cowboy,  
overly elegant, code ego,  
real soon now, stress  
difficult to talk to

8-5, teamwork,  
code reviews,  
communication,  
coding standards,  
snacks, communal code  
tests, tests, tests,  
maintainable, documented  
predictable,  
copes with change

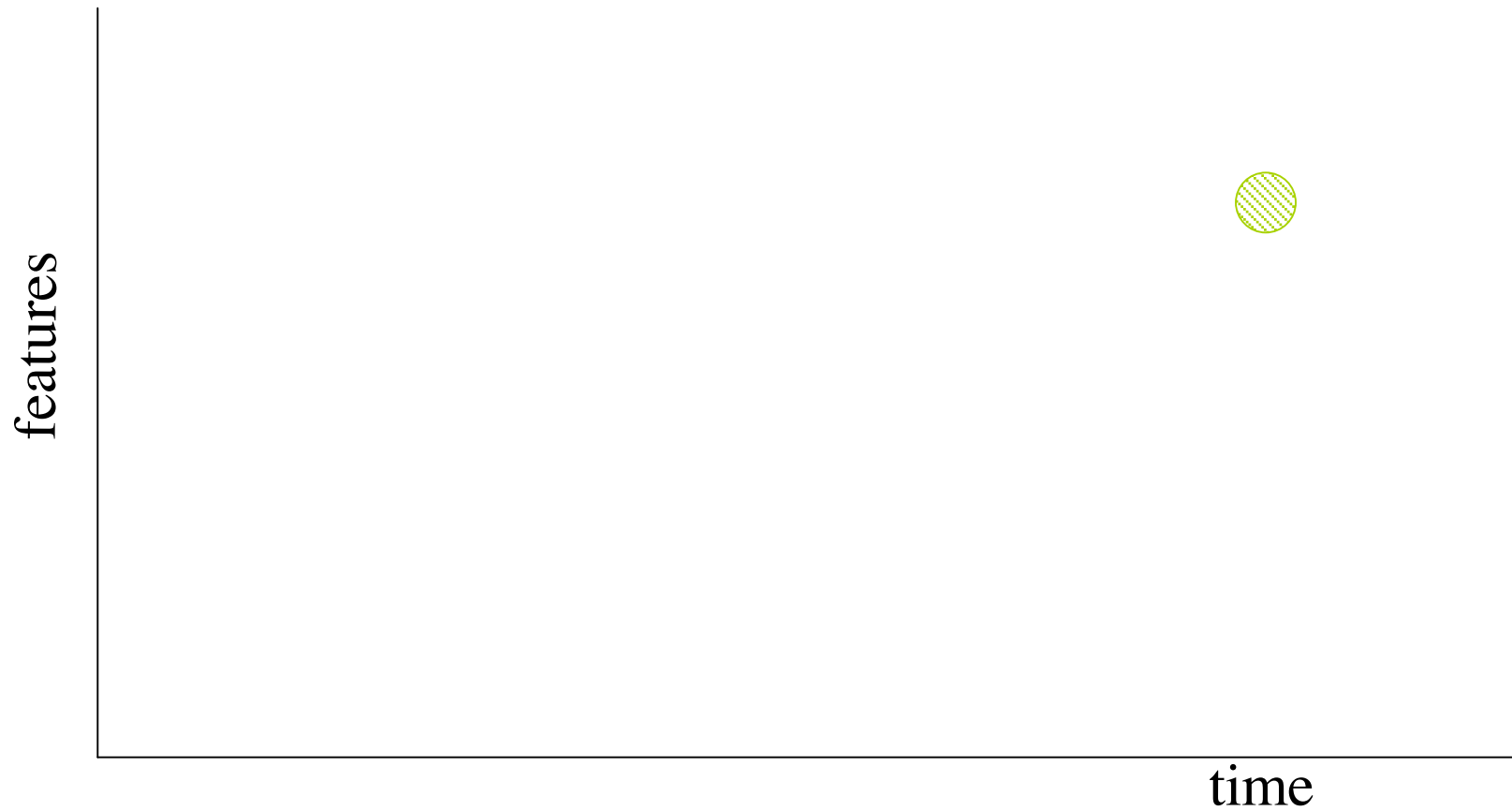
# XP Practices

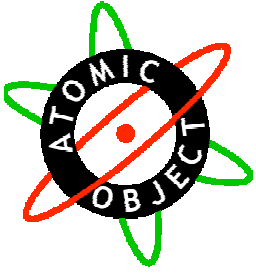
Ron Jeffries, XProgramming.com



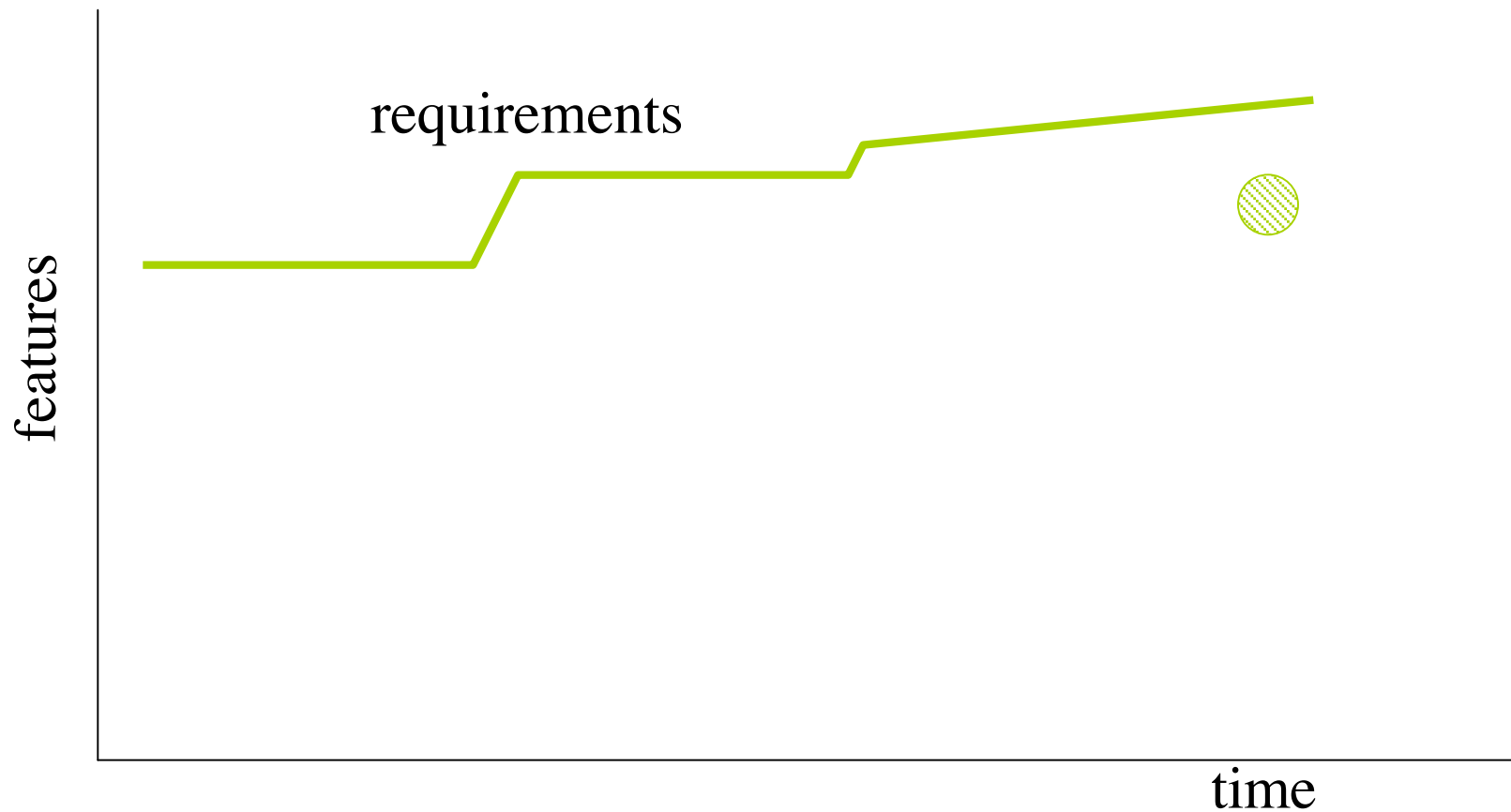


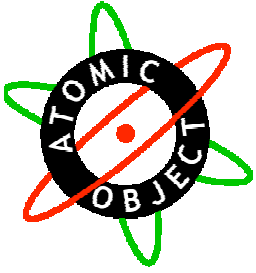
# XP planned & reality



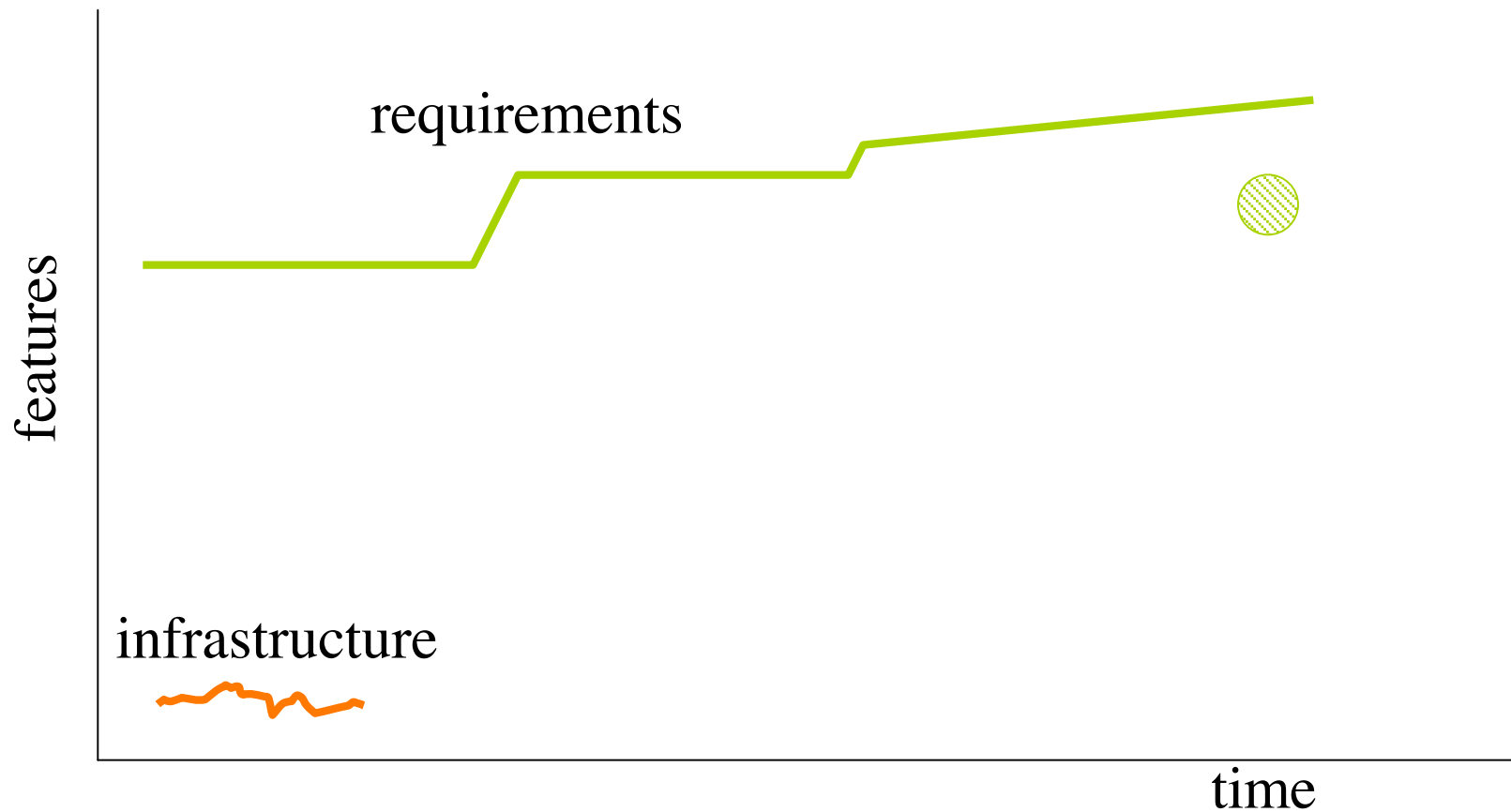


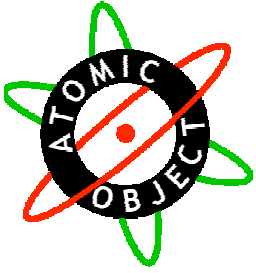
# XP planned & reality



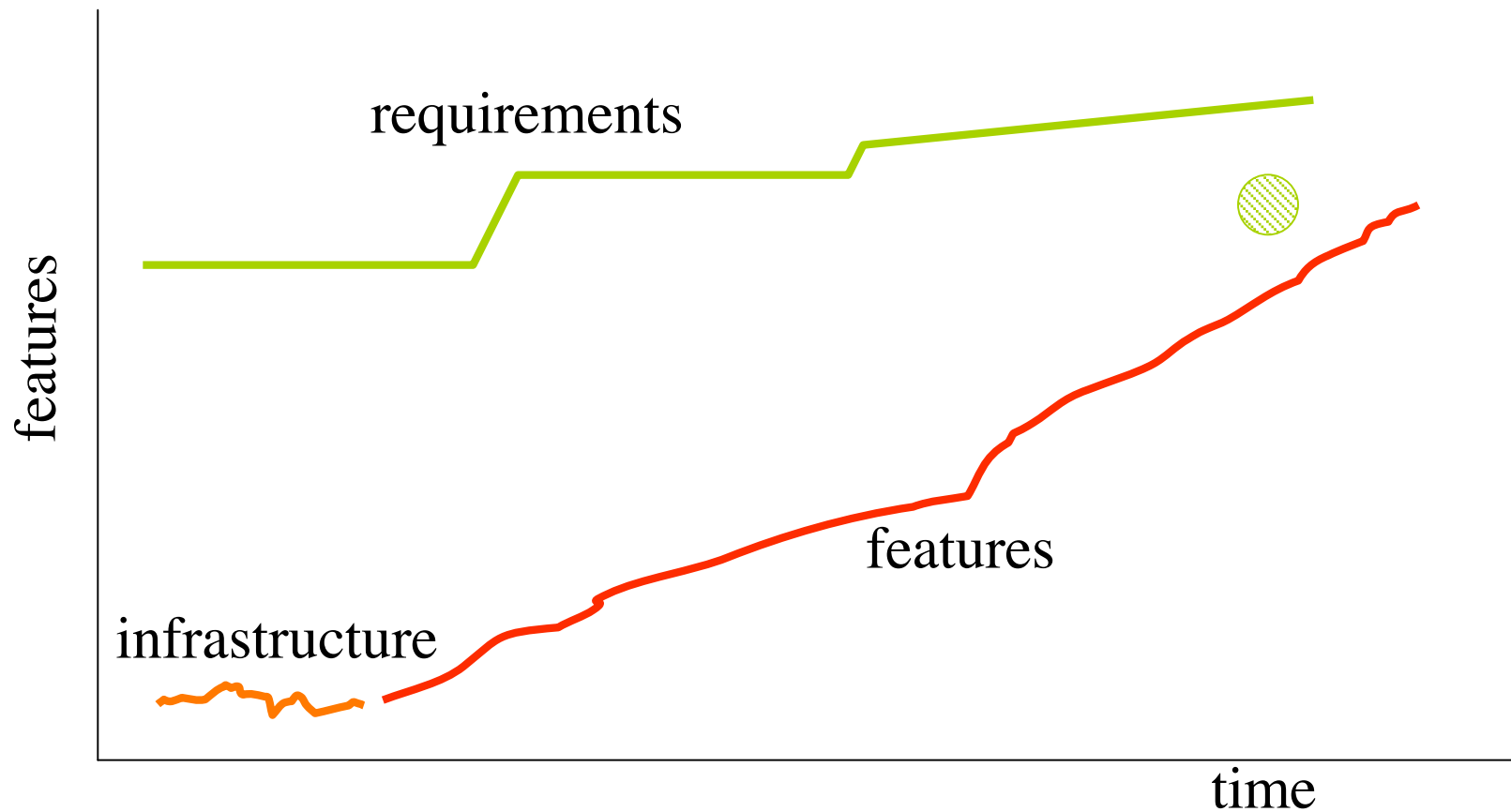


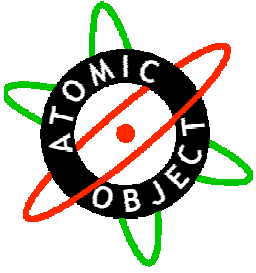
# XP planned & reality



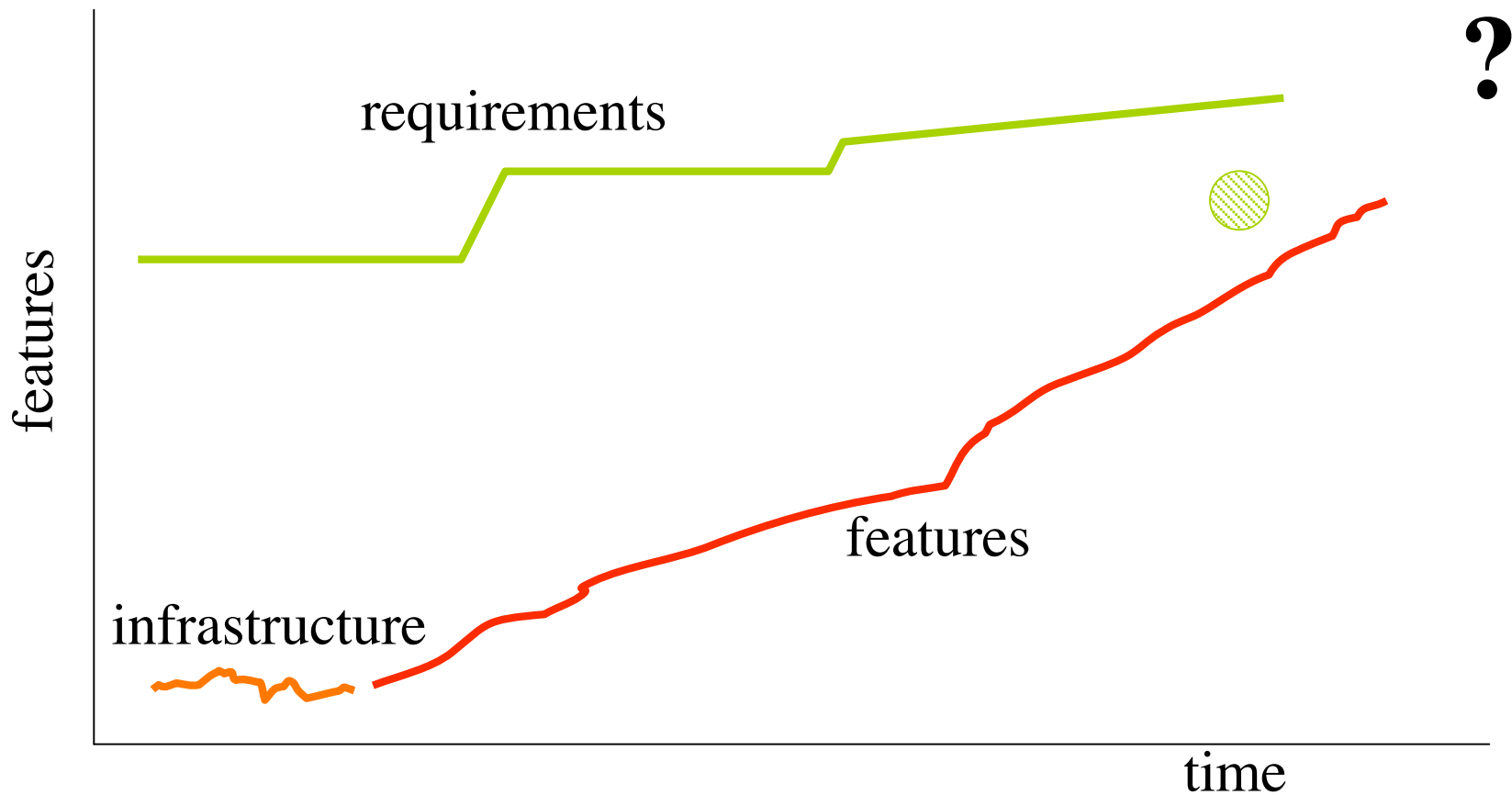


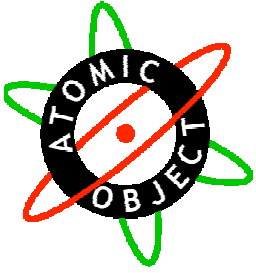
# XP planned & reality





# XP planned & reality





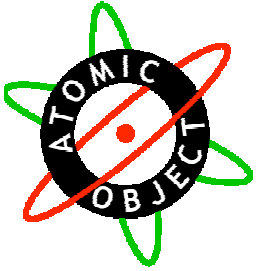
# Data-driven Projects

How do project managers know the state of a project?

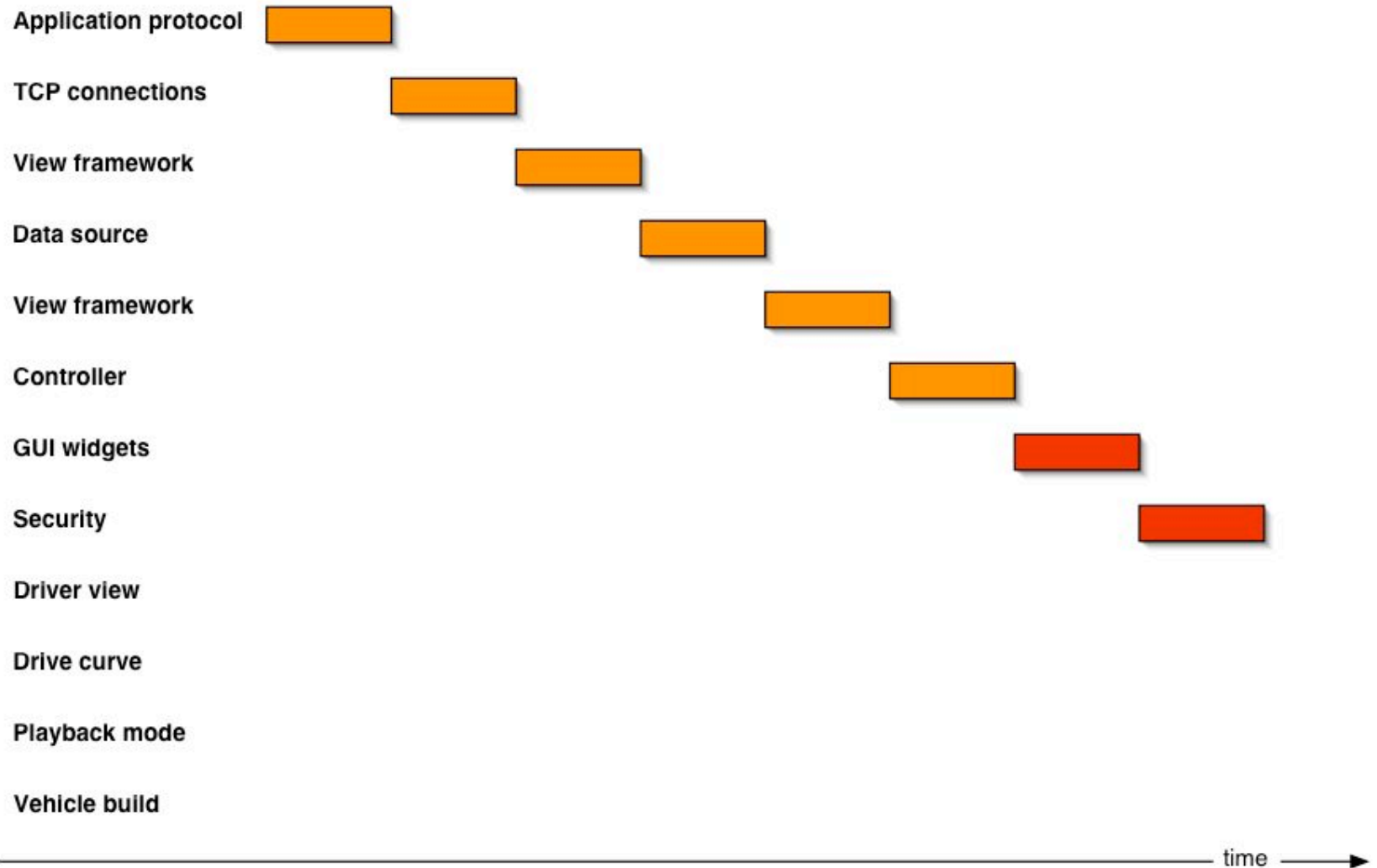
How can they predict delivery date?

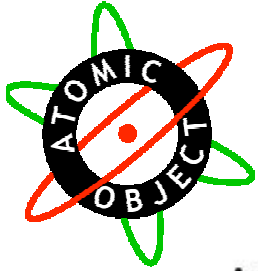
How well does a project cope with change?



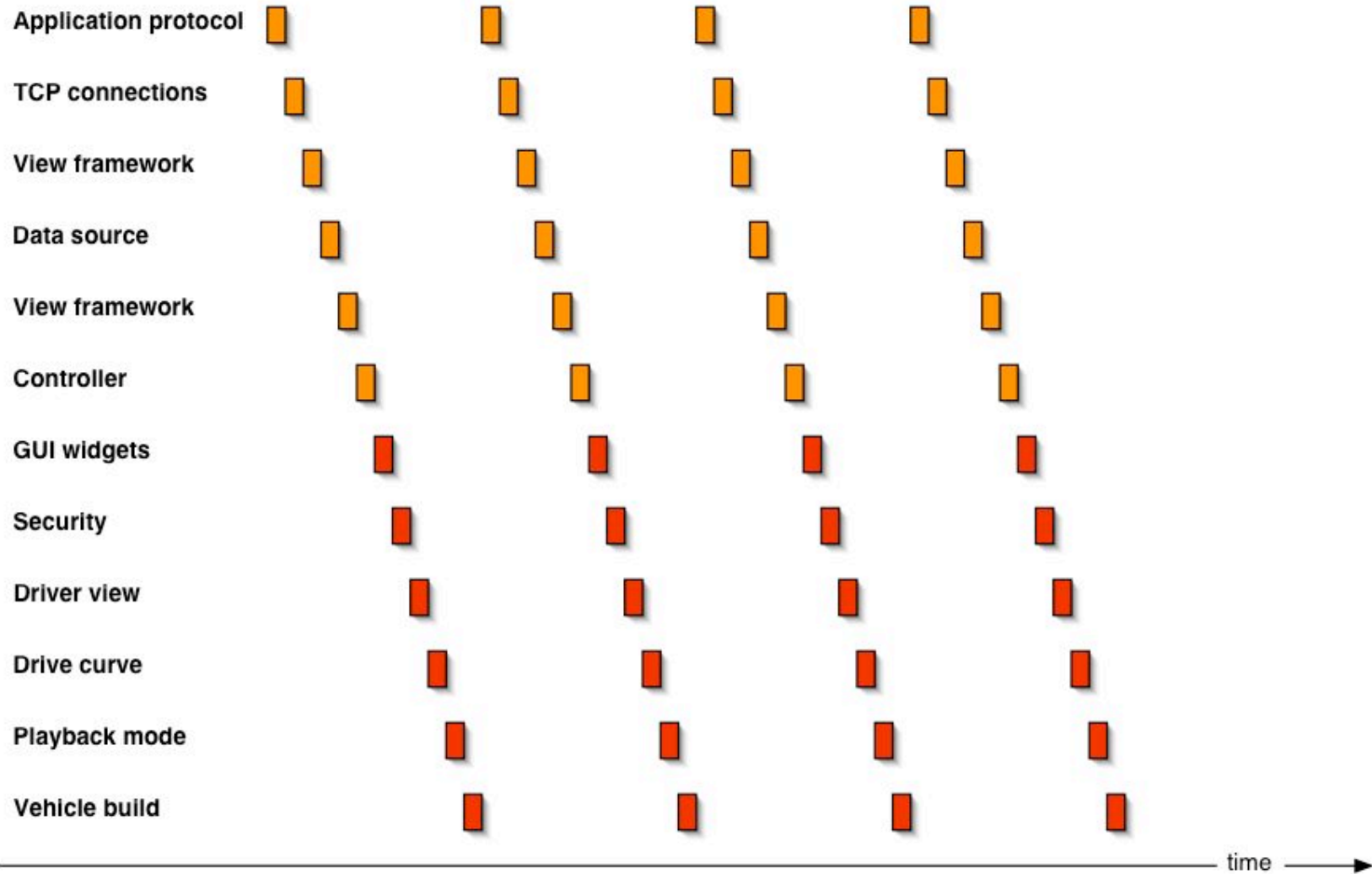


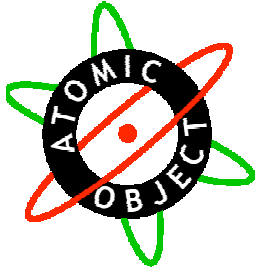
# Infrastructure first



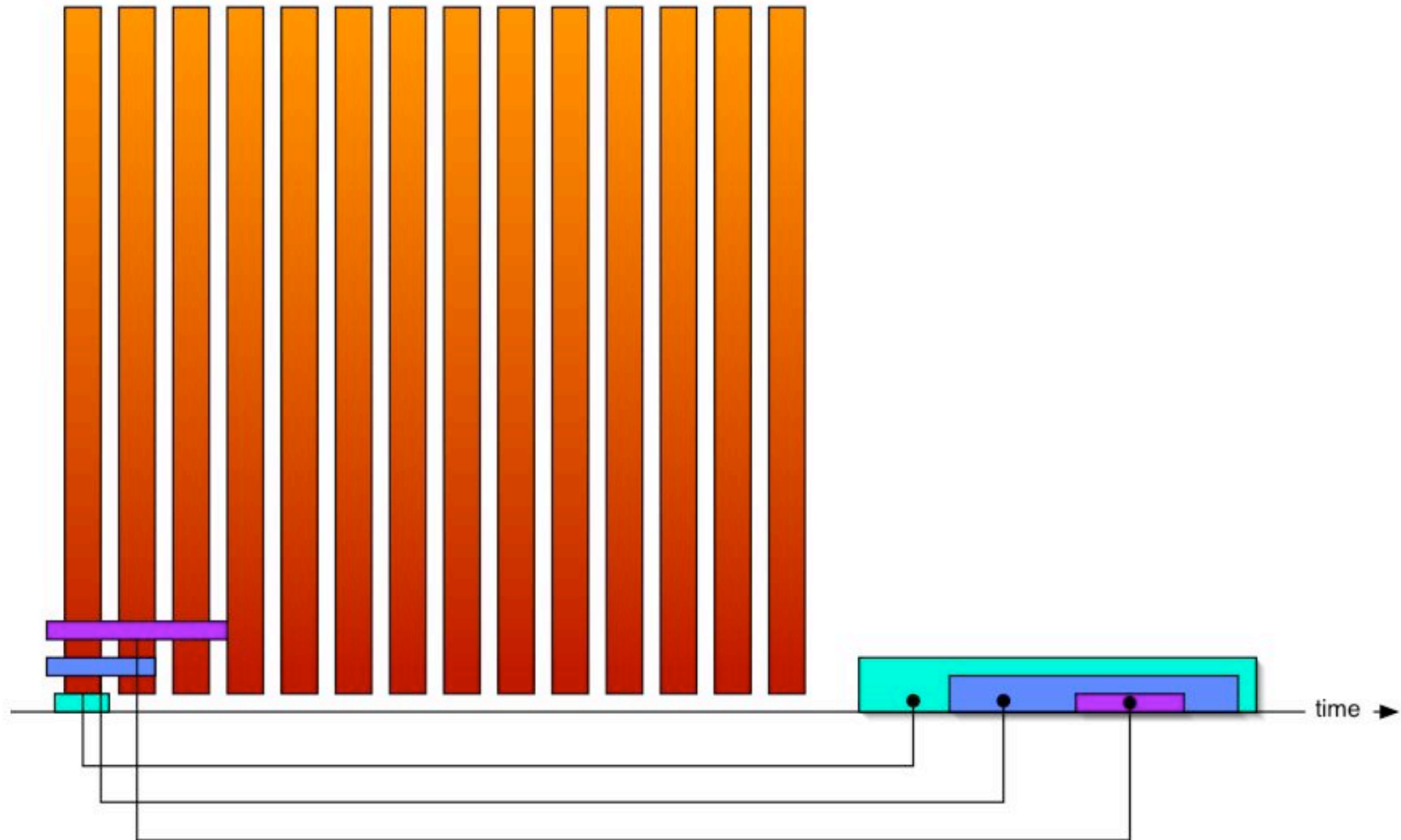


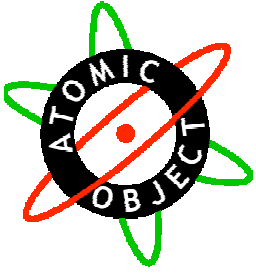
# Features first



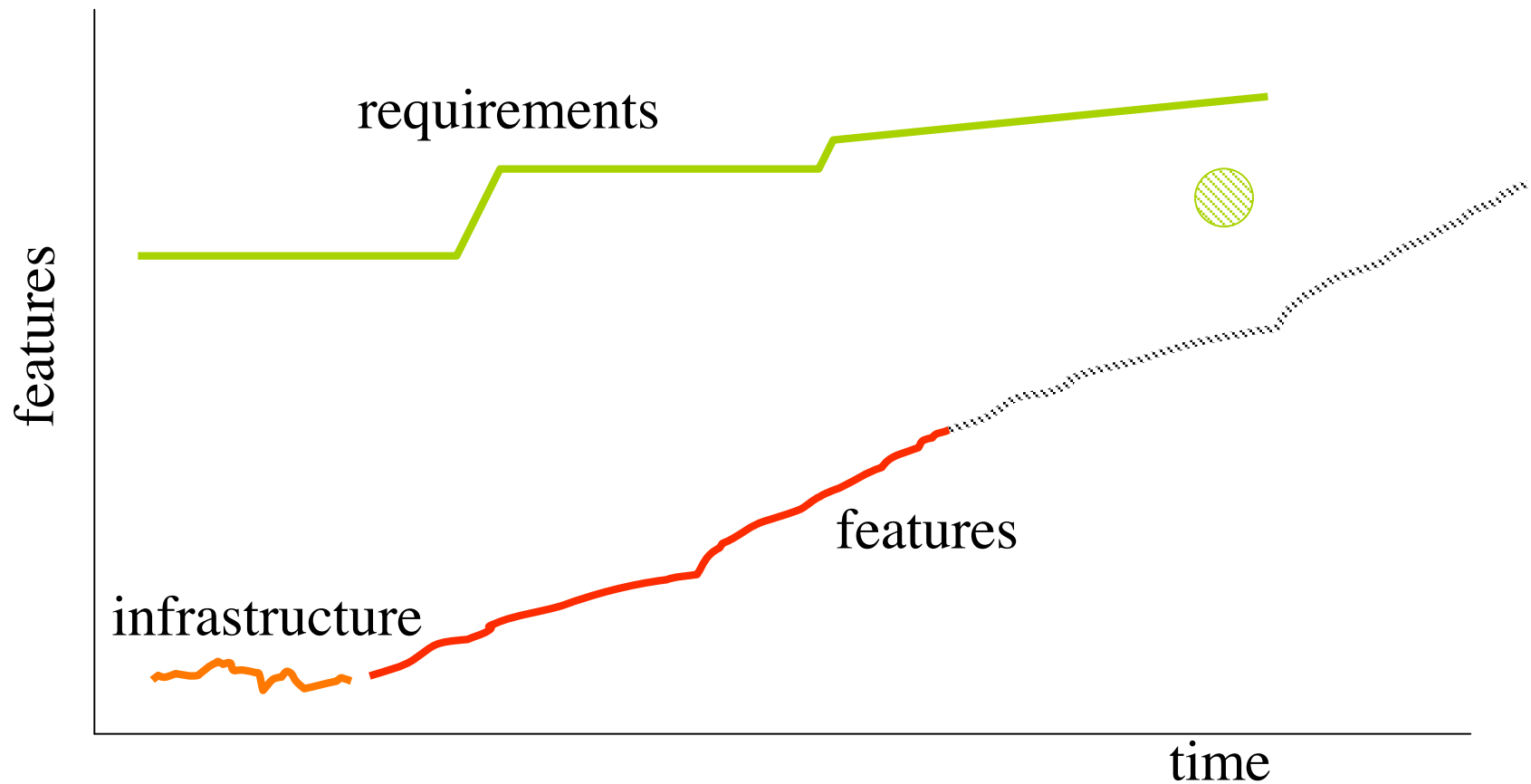


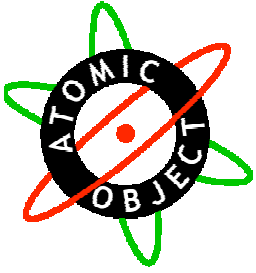
# Measure and Predict





# Data-driven project management





# Can this possibly work?

- Resisting the lure of infrastructure
- Continuously working system
- Frequent integration
- Refactoring
- Test driven development
- Build and test automation