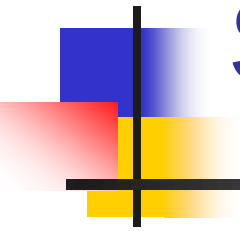
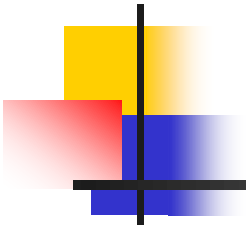


# SWE 205 - Introduction to Software Engineering

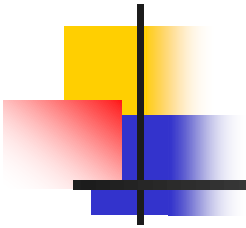
Lecture 3

Introduction to Software Engineering



# Lecture Objectives

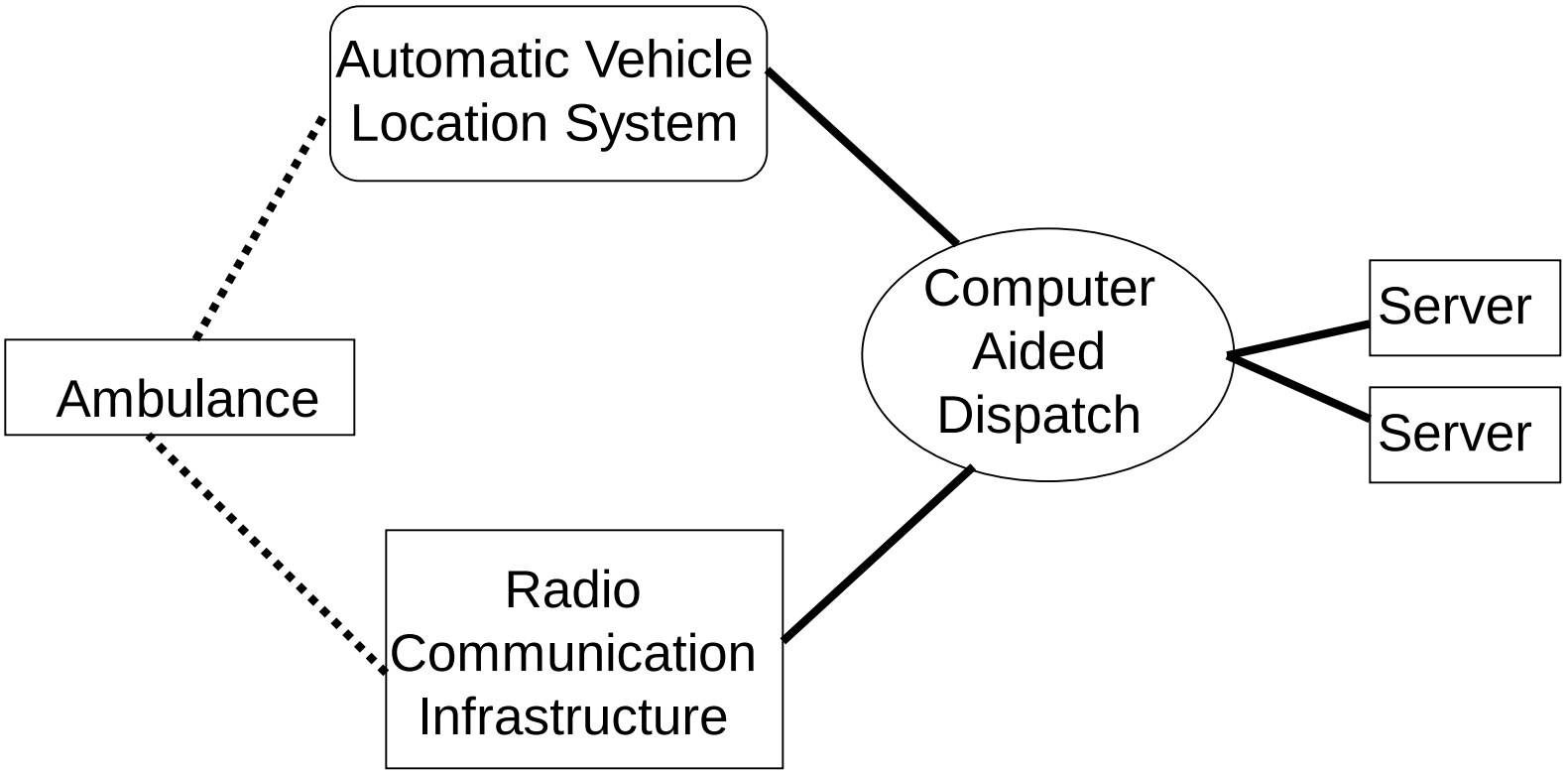
- Software Myths.
- Software Engineering Challenges.
- What is a software process?
- Generic framework activities.

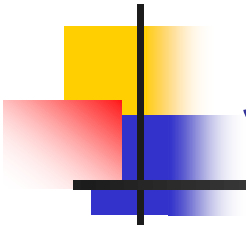


# Important Question

- Why do we continue to have difficulty in software development projects?

# London Ambulance Service Case Study





# Software Myths

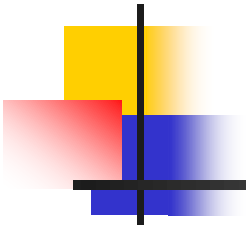
- Affect managers, stakeholders, and practitioners.
- Are believable because they often have elements of truth.

*but…*

- Invariably lead to bad decisions,

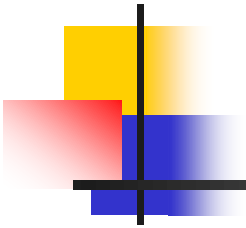
*therefore….*

- Insist on reality as you navigate your way through software engineering.



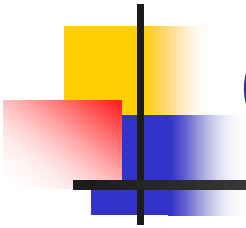
# Management Myths

- *‘We already have books full of standards and procedures for building software. That will provide my people with everything they need to know’.*
- *‘My people do have state-of-the-art software development tools. After all, we buy them the latest computers’.*



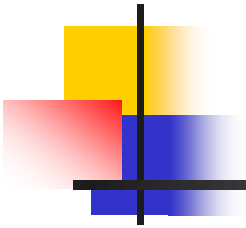
# Management Myths

- *‘If we get behind schedule we can add more programmers and catch up’.*
- *‘If I decide to outsource the software project to a third party, I can just relax and let that firm build it.*



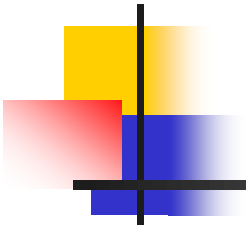
# Customer Myths

- ‘A general statement of objectives is sufficient to begin writing software - we can fill in the details later’.
- ‘Project requirements continually change but change can be easily accommodated because software is flexible’.



# Practitioner's Myths

- 'Once we write the program and get it to work our job is done'.
- 'Until I get the program running I really have no way of assessing its quality'.
- 'The only deliverable for a successful project is the working program'.

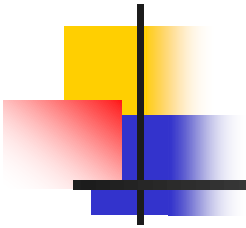


# Key Challenges

- Heterogeneity
  - Developing techniques for building software that can cope with heterogeneous platforms and execution environments;
- Delivery
  - Developing techniques that lead to faster delivery of software;
- Trust
  - Developing techniques that demonstrate that software can be trusted by its users.

# Key Challenges

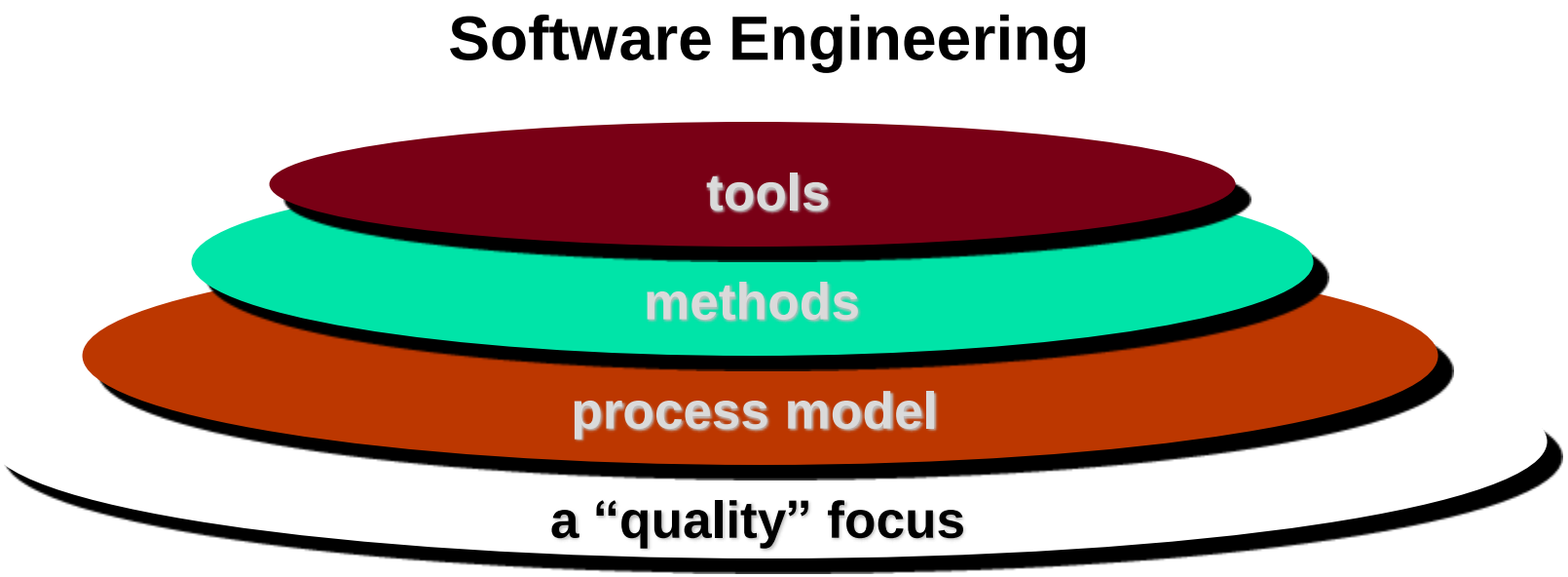
- An accompanying shift from a concern with *whether* a system will work towards how *well it* will work.
- Components are selected and purchased ‘off the shelf’ (COTS) with development effort being refocused on configuration and interoperability.



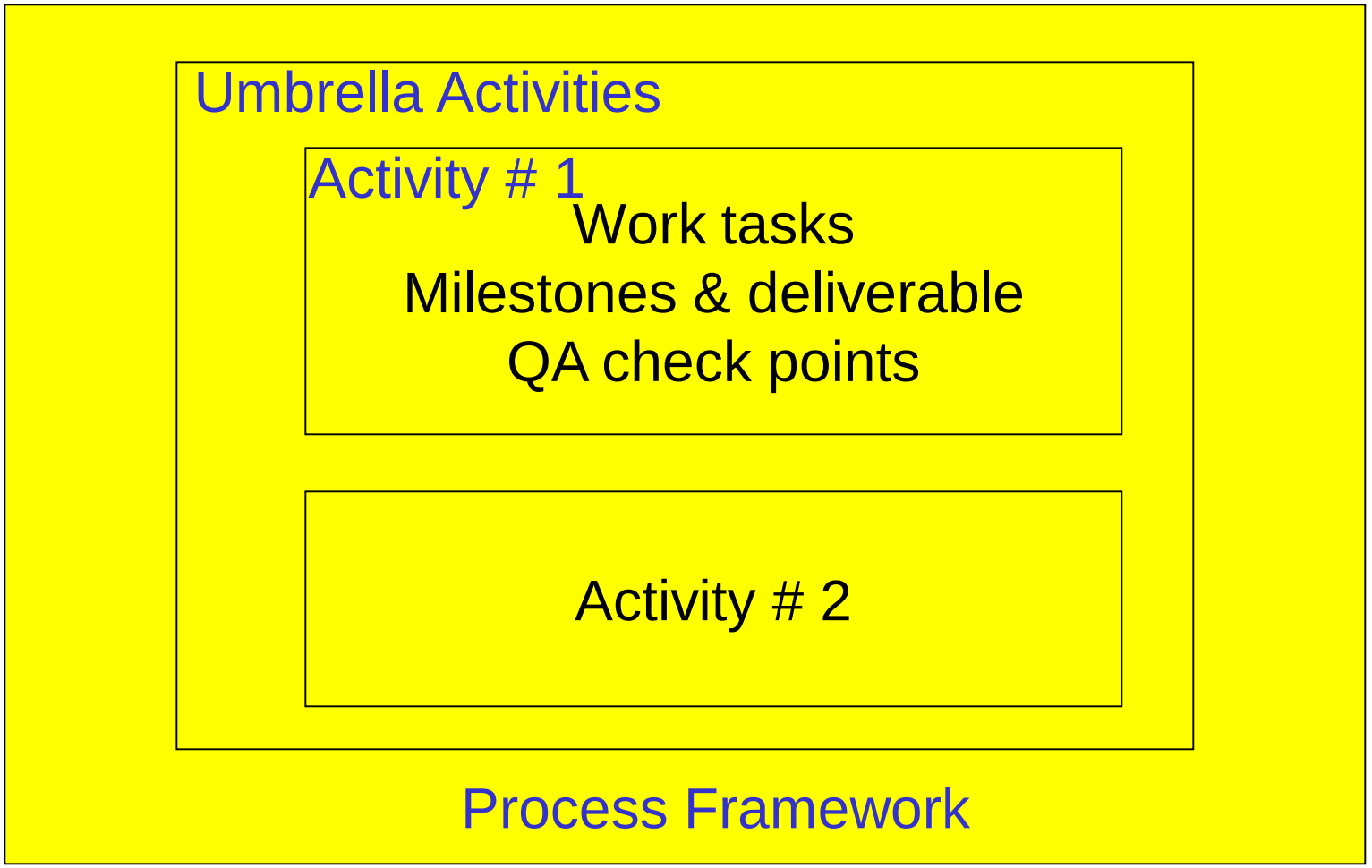
# How a Project Starts?

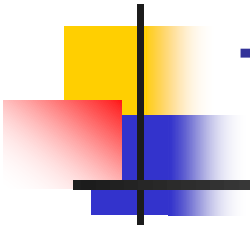
- Every software project is precipitated by some business need
  - Need to correct a defect in an existing application.
  - Need to adapt a legacy system to a changing business environment.
  - Need to extend the functions and features of an existing application.
  - Need to create a new product or system.

# A Layered Technology



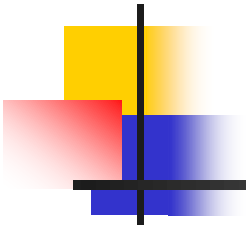
# A Software Process





# The software process

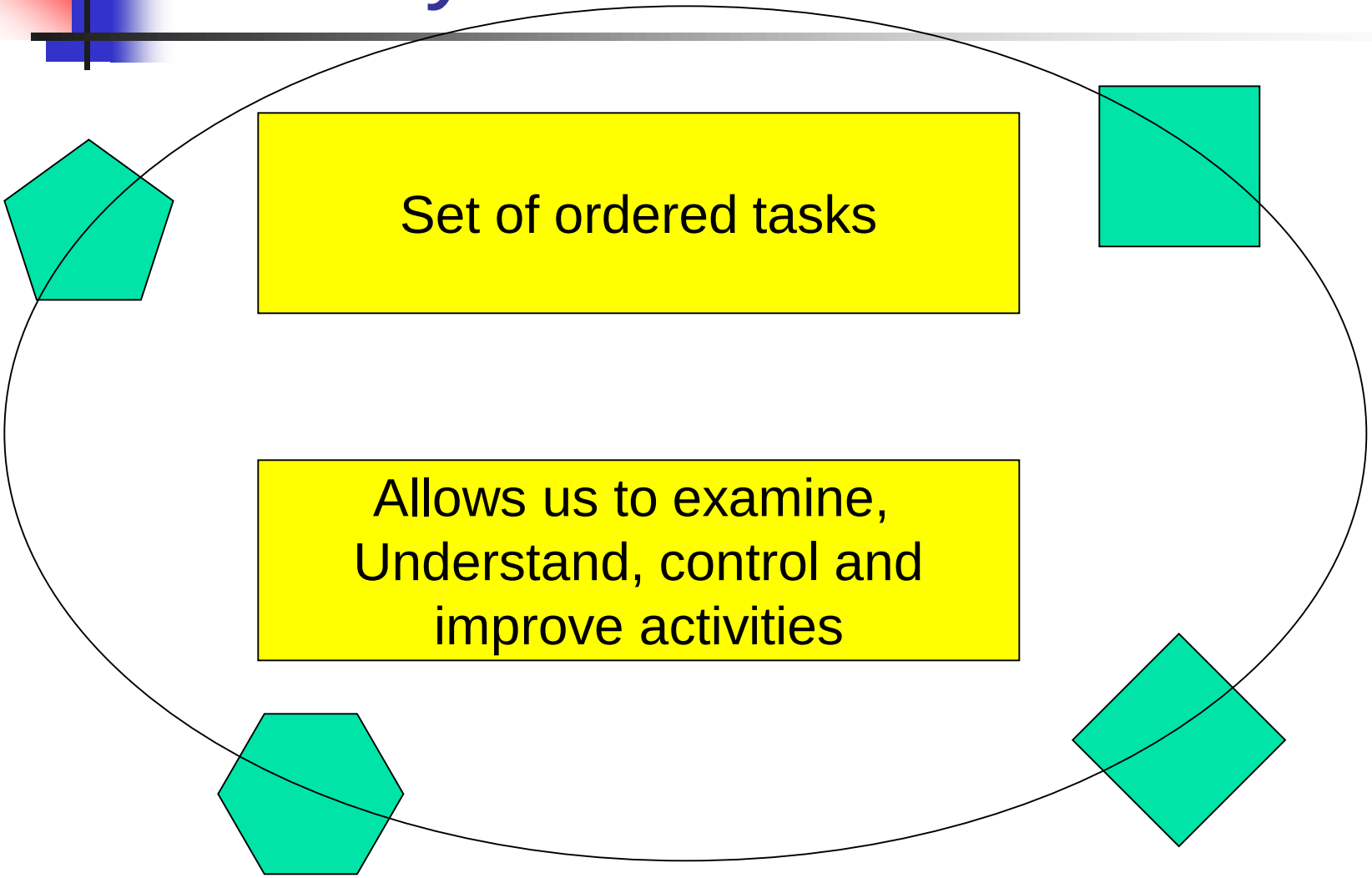
- A structured set of activities required to develop a software system
  - Specification;
  - Design;
  - Validation;
  - Evolution.

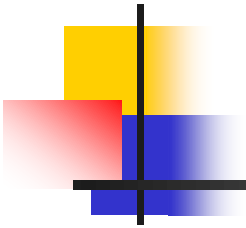


# Umbrella Activities

- Software project management.
- Formal technical reviews.
- Software quality assurance.
- Measurements.
- Risk management.

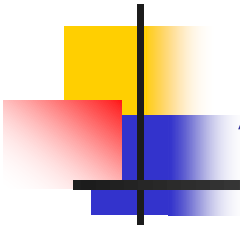
# Software Process - Software Life Cycle





# Key Points

- Approaches which work for constructing small programs for personal use do not scale-up to the challenges of real software construction.
- Software processes are the activities involved in producing and evolving a software system.



# Announcements

- Quiz 1 moved from
  - Saturday 18/10/2008 to
  - Wednesday 22/10/2008.
- Project Team member names by Wednesday 22/10/2008.
  - A team will be of four students.