

Software Project Management

Course Contents:

Unit 04: Modern Approach to Software Project and Economics

- 3.1 Elements of Modern Software Projects and Management Principles
- 3.2 Next-Generation Software Economics and Cost Models
- 3.3 Modern Process Transition – Paradigm Shifts

Modern Software Development Project

- In short, software development is the overall process involved when taking a software project from beginning to production delivery.
- This process incorporates the design, documentation, programming, testing and ongoing maintenance of a software deliverable.

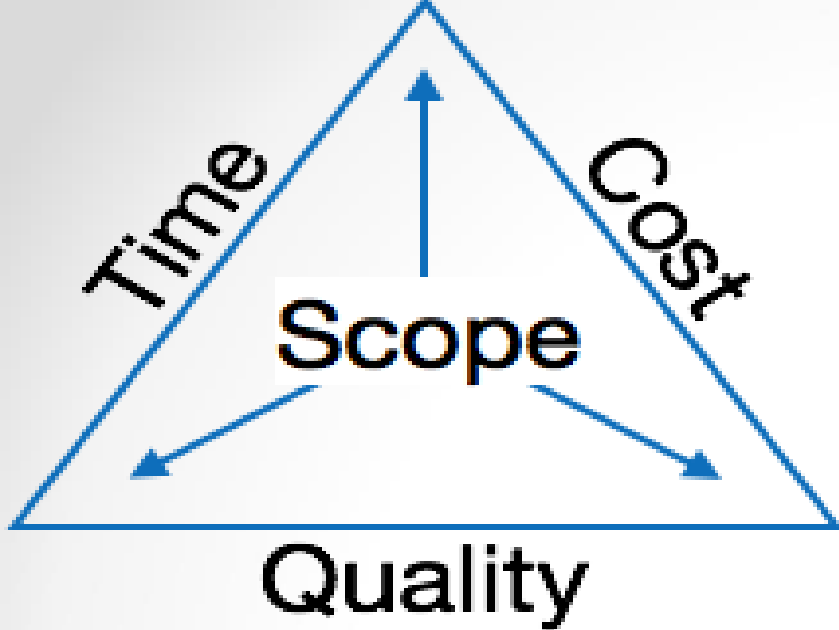
Modern Software Project Management

- Modern S/W Project Management that makes this a perfect time to reflect on what modern Project Management has become and why it is such a critical issue for business organizations of all types.
- Project Management is a discipline for planning, leading, organizing and controlling a well-defined collection of work.

Modern Software Project Management

- Software is said to be an intangible product.
- Software development is a kind of all new stream in world business and there's very little experience in building software products.
- Most software products are adapt/modify made to fit client's requirements.

- The most important is that the underlying technology changes and advances so frequently and rapidly that experience of one product may not be applied to the other one.
- All such business and environmental constraints bring risk in software development hence it is essential to manage software projects efficiently.



- The image above shows triple constraints for software projects.
- It is an essential part of software organization to deliver quality product, keeping the cost within client's budget constrain and deliver the project as per scheduled.

- There are several factors, both internal and external, which may impact this triple constrain triangle.
- Therefore, software project management is essential to incorporate user requirements along with budget and time constraints.

Five Essential Elements for Successful Software Development

- Integrated Development Environment (IDE).
- Source Control.
- Automated Testing.
- Automated Build.
- Defect Management.

The 5 Most Important Elements of Successful Project Management

- Have Clear Project Goals. Make sure you have all the details in front of you before you start.
- Be Dynamic. Once you have your plan in place, remain flexible.
- Communication. You need to ensure your team clearly communicates with one another.
- Stay on Track.
- Review The Project to Improve for The Next Time

Management Activities



Planning the
Works



Measure the
Project



Manage the
Risk



Resource
Planning



Reuse



Ensure Quality



Communicate with
Stakeholders



Manage the Project Team

Top 10 Management Principles of Iterative Development

- Base the process on an **architecture-first** approach.
- Establish an **iterative lifecycle process** that confronts risk early.
- Transition design methods to emphasize **component-based development**.
- Establish a **change-management environment**.
- Enhance **change freedom** through tools.
- Capture design artifacts in rigorous, **model-based notation**.

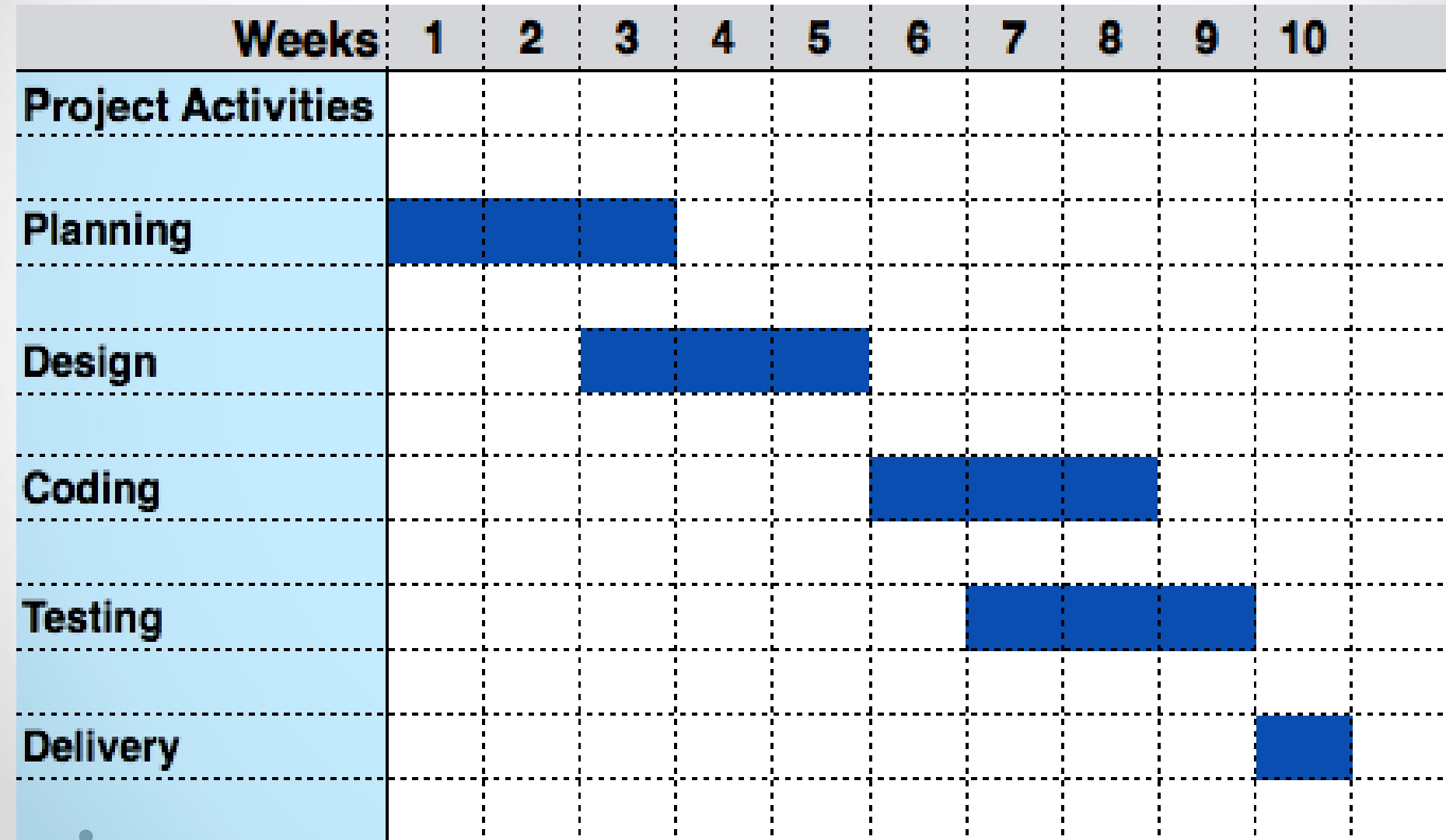
- Instrument the process for **objective quality control and progress assessment**.
- Use a **demonstration-based approach** to assess intermediate artifacts.
- Plan intermediate releases in groups of usage scenarios with **evolving levels of detail**.
- Establish a **configurable process** that is economically scalable.

Project management principles are necessary assets when charting a path to completion.

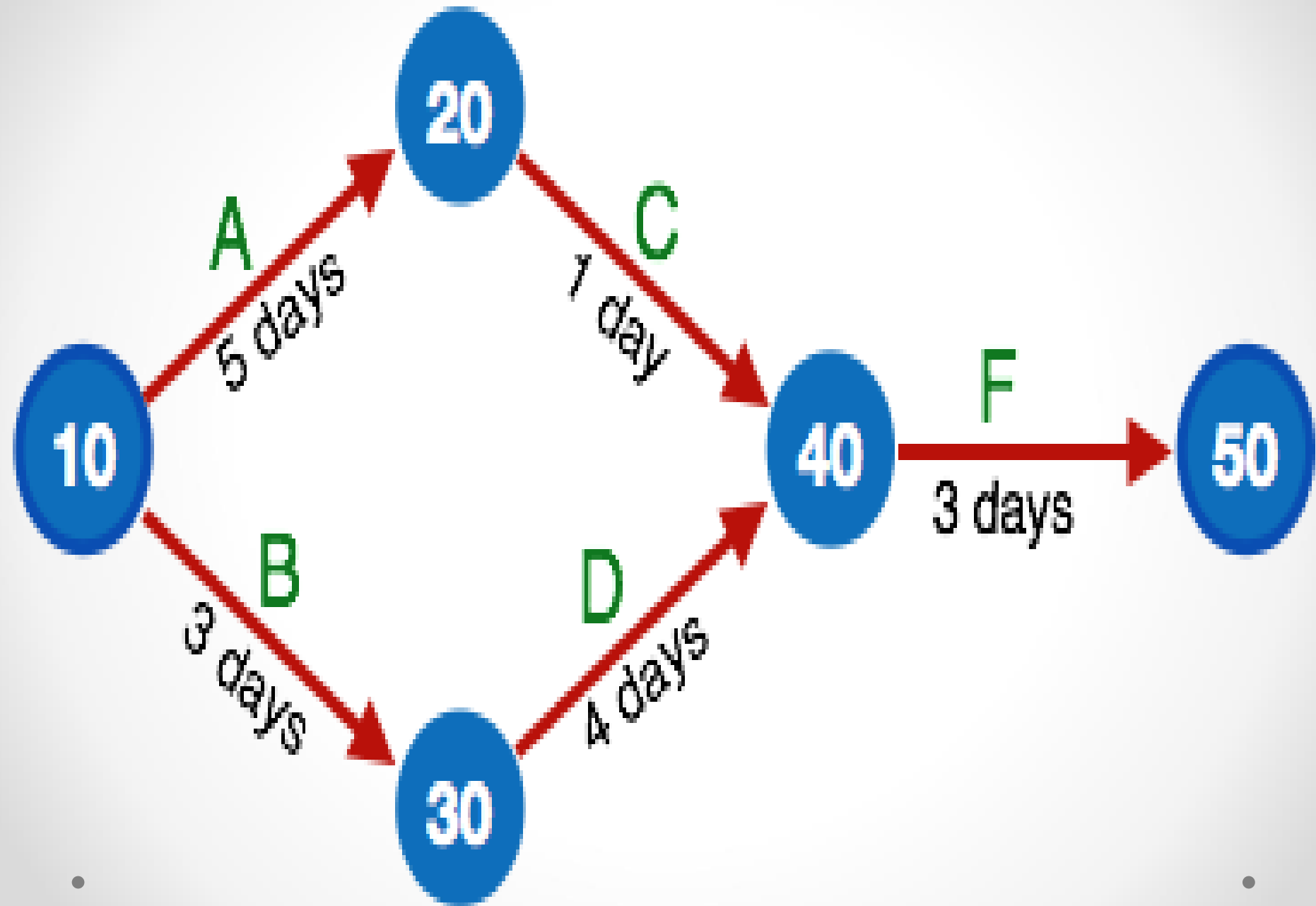
- Project structure
- Definition phase
- Clear goals
- Transparency about project status
- Risk recognition
- Managing project disturbances
- Responsibility of the Project Manager
- Project success

Project Management Tools

1. Gantt Chart

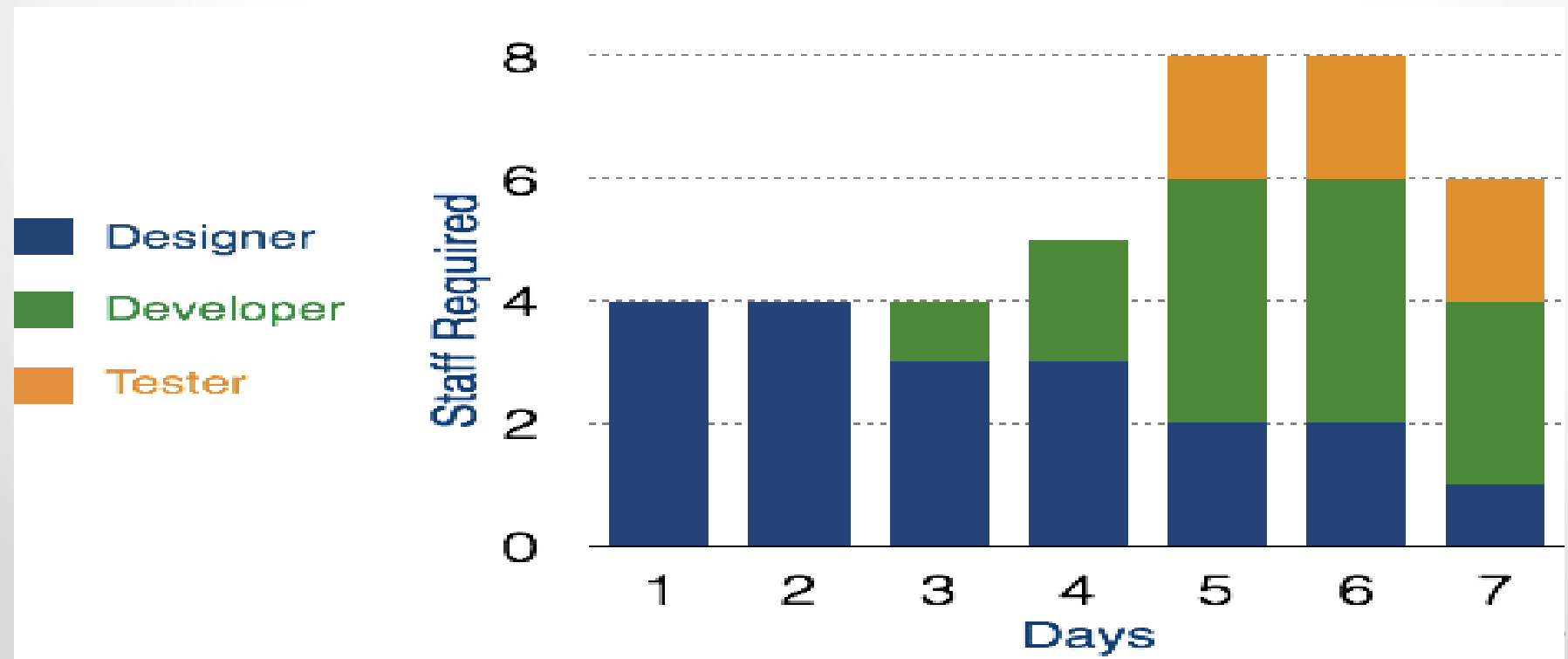


2. PERT Chart



3. Resource Histogram

Staff	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
Designer	4	4	3	3	2	2	1
Developer	0	0	1	2	4	4	3
Tester	0	0	0	0	2	2	2
Total	4	4	4	5	8	8	6



Software Management Best Practices

There is nine best practices:

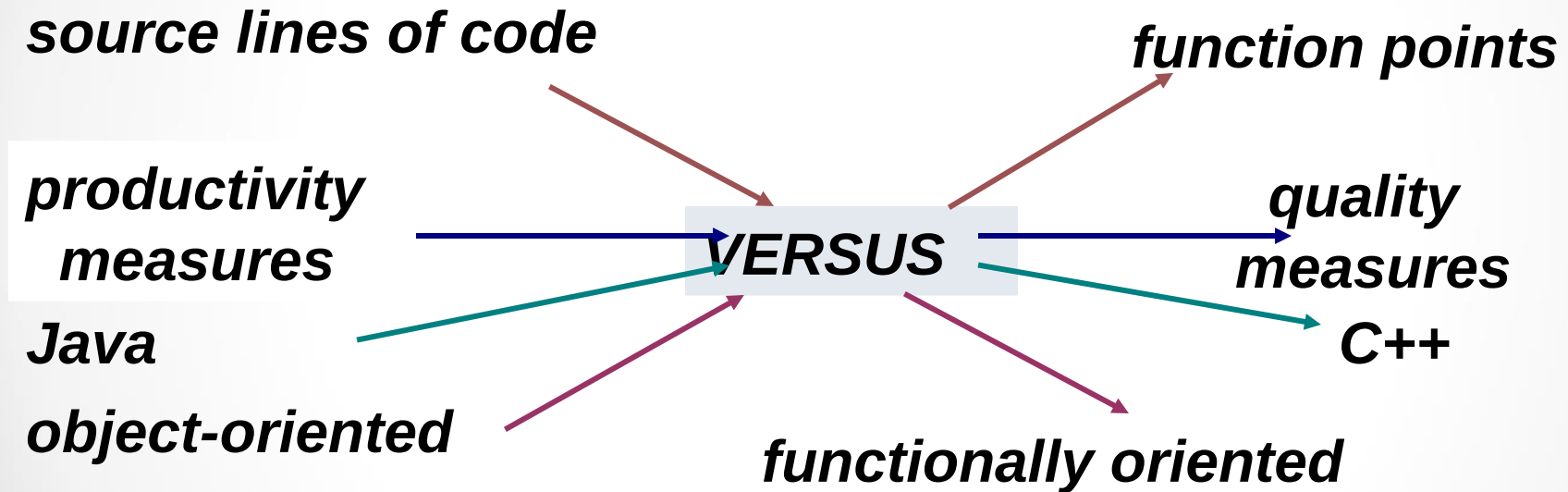
1. Formal risk management
2. Agreement on interfaces
3. Formal inspections
4. Metric-based scheduling and management
5. Binary quality gates at the inch- pebble level

6. Program-wide visibility of progress versus plan
7. Defect tracking against quality targets
8. Configuration management
9. People-aware management accountability

Next-Generation Software Economics & Cost Models

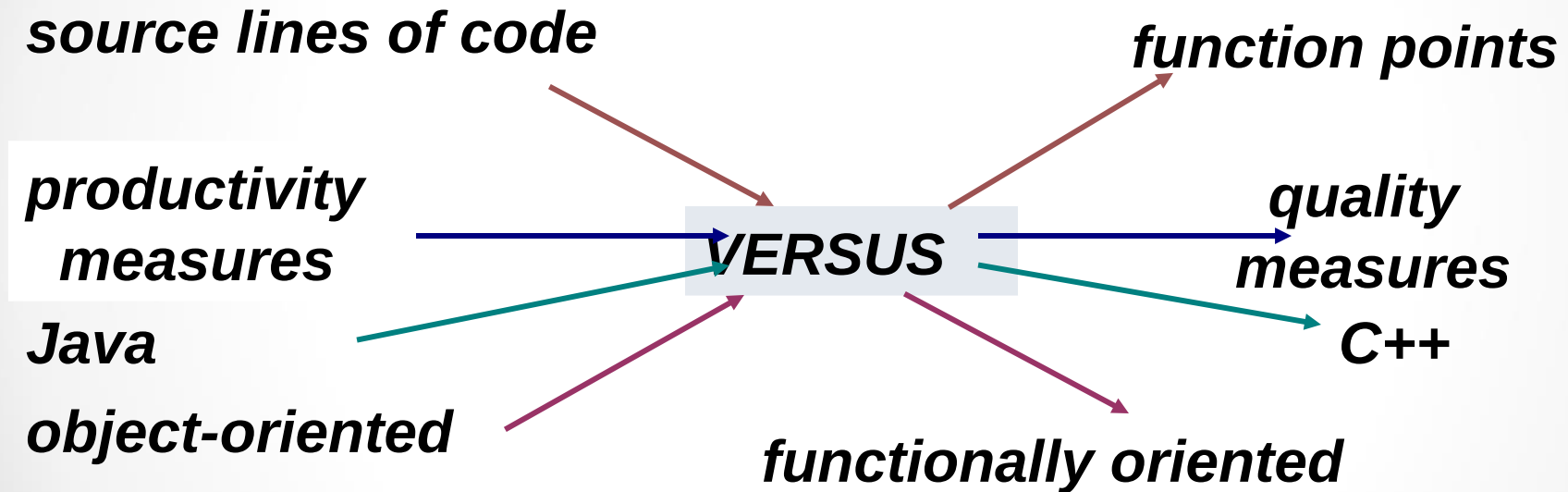
Next-Generation Cost Models

- Software experts hold widely different opinions about software economics and its symptom in software cost estimation models:



Next-Generation Cost Models

- ❑ Software experts hold widely different opinions about software economics and its symptom in software cost estimation models:



- ❑ It will be difficult to improve based on estimation models while the project data going into these models are highly lacking a mutual relationship, and are based on differing process and technology foundations.

- ❑ Some of today's popular software cost models are not well matched to an computational software process focused an architecture-first approach
- ❑ Many cost estimators are still using a traditional or common process experience base to estimate a modern project profile
- ❑ Then, a next-generation software cost model should clearly separate architectural engineering from application production, just as an architecture-first process does.

❑ There are two major improvements in next-generation software cost estimation models:-

- Separation of the engineering stage from the production stage: which will force estimators to differentiate between architectural scale and implementation size.
- Detailed design notations such as UML will offer an opportunity to define units of measure for scale that are more standardized and therefore can be automated and tracked.

Modern Software Economics

❑ Changes: that provide a good description of what an organizational manager should attempt for in making the transition to a modern process:

1. Finding and fixing: a software problem after delivery costs is 100 times more than fixing the problem in early design phases.
2. We can compress software development schedules 25% of nominal, but no more.
3. For every \$100 you spend on development, you will spend \$200 on maintenance.
4. Software development and maintenance costs are primarily a function of the number of source lines of code.

5. Variations among people account for the biggest differences in software productivity.
6. Only about 15% of software development effort is devoted to programming.
7. Software systems and products cost 3 times as much per SLOC as individual software programs.
8. 80% of the contribution comes from 20% of the contributors.

Remove Project Obstacles

- Daily Project Meeting Identifies
 - ◆ Bottlenecks
 - ◆ Needed resources
 - ◆ Issues that need resolving
 - ◆ Conflicts between stakeholders
 - ◆ Differences between plans and actualities
 - ◆ Processes that need improvement



Why Do Software Projects Fail?

- ♦ We fail to properly manage risks



- ♦ We don't build the right thing

Question

- ♦ We are blindsided by technology

Notice the “**We**”. As project managers, we develop idealistic plans, we set unrealistic schedules, we deceive ourselves and others, and we refuse to face reality. These projects eventually enter “free fall” with no one taking responsibility and everyone waiting for the crash (while sending out resumes).

Two Types of Risk

- Project and Process Risk



- ♦ What could go wrong with this project?

Define

- Product or Requirements Risk

- ♦ Which faults would be most damaging to the stakeholders?

What Do Project Managers Do?



Direct

- Team Management
- Plan, Schedule, Track
- Resource Allocation
- Project Direction
- Politics
- Remove Project Obstacles

Assignment 04:

- **Discussed in following:**
 - 1. Modern Process Transitions.**
 - 2. Denouement**

Modern Process Transitions

Culture Shifts

- ❑ Several culture shifts must be overcome to transition successfully to a modern software management process:

Modern Process Transitions

Culture Shifts

- ❑ Several culture shifts must be overcome to transition successfully to a modern software management process:
 - Lower level and mid-level managers are performers
 - Requirements and designs are fluid and tangible
 - Good and bad project performance is much more obvious earlier in the life cycle
 - Artifacts are less important early, more important later

- Real issues are surfaced and resolved systematically
- Quality assurance is everyone's job, not a separate discipline
- Performance issues arise early in the life cycle
- Investments in automation is necessary
- Good software organization should be more profitable

Denouement

- ❑ Good way to transition to a more mature iterative development process that supports automation technologies and modern architectures is to take the following shot:

Denouement

❑ Good way to transition to a more mature iterative development process that supports automation technologies and modern architectures is to take the following shot:

➤ Ready.

Do your homework. Analyze modern approaches and technologies. Define your process. Support it with mature environments, tools, and components. Plan thoroughly.

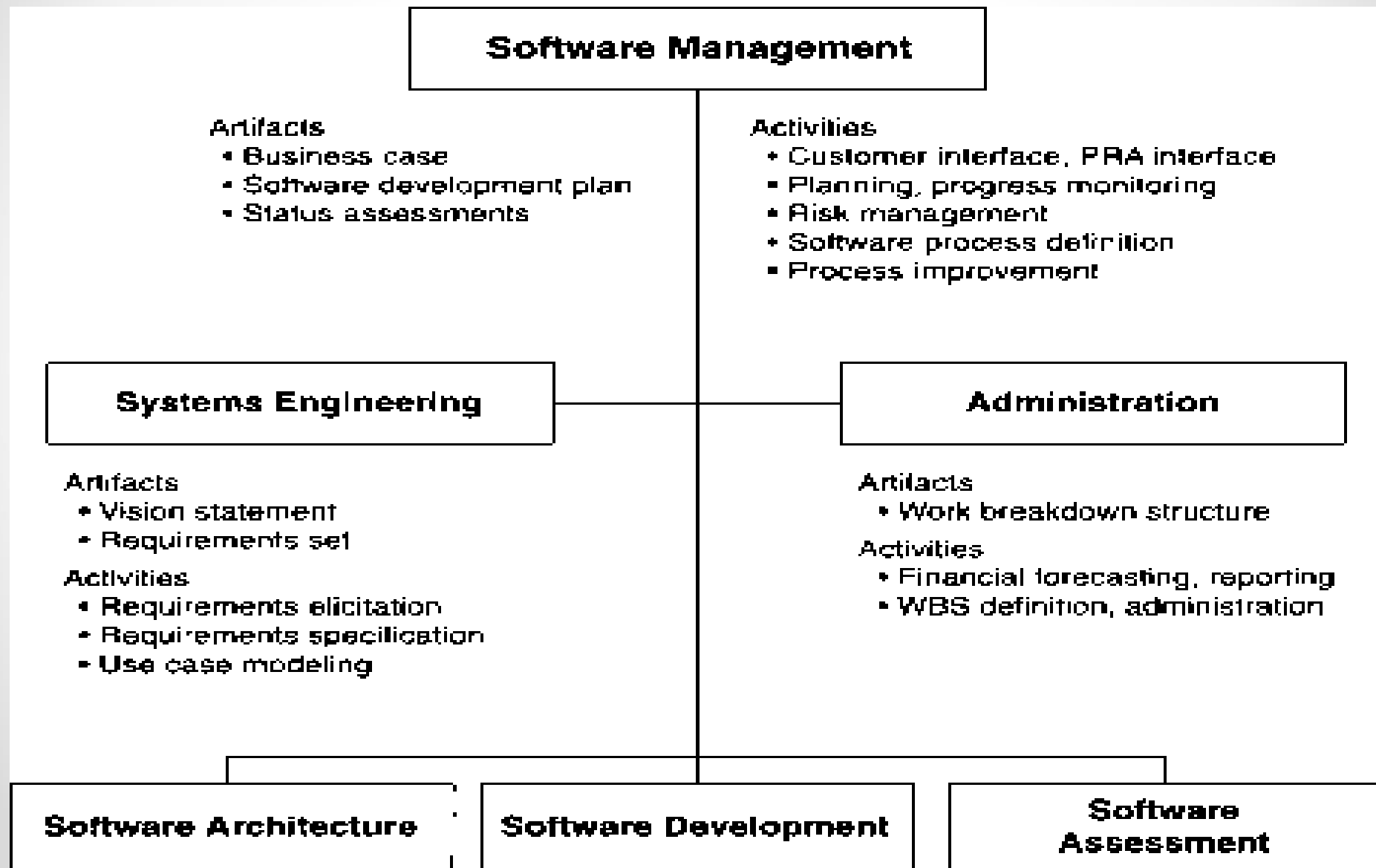
➤ Aim.

Select a critical project. Staff it with the right team of complementary resources and demand improved results.

➤ Fire.

Execute the organizational and project-level plans with vigor and follow-through.

- This structure can be tailored to the size and circumstances of the specific project organization.



Thank you