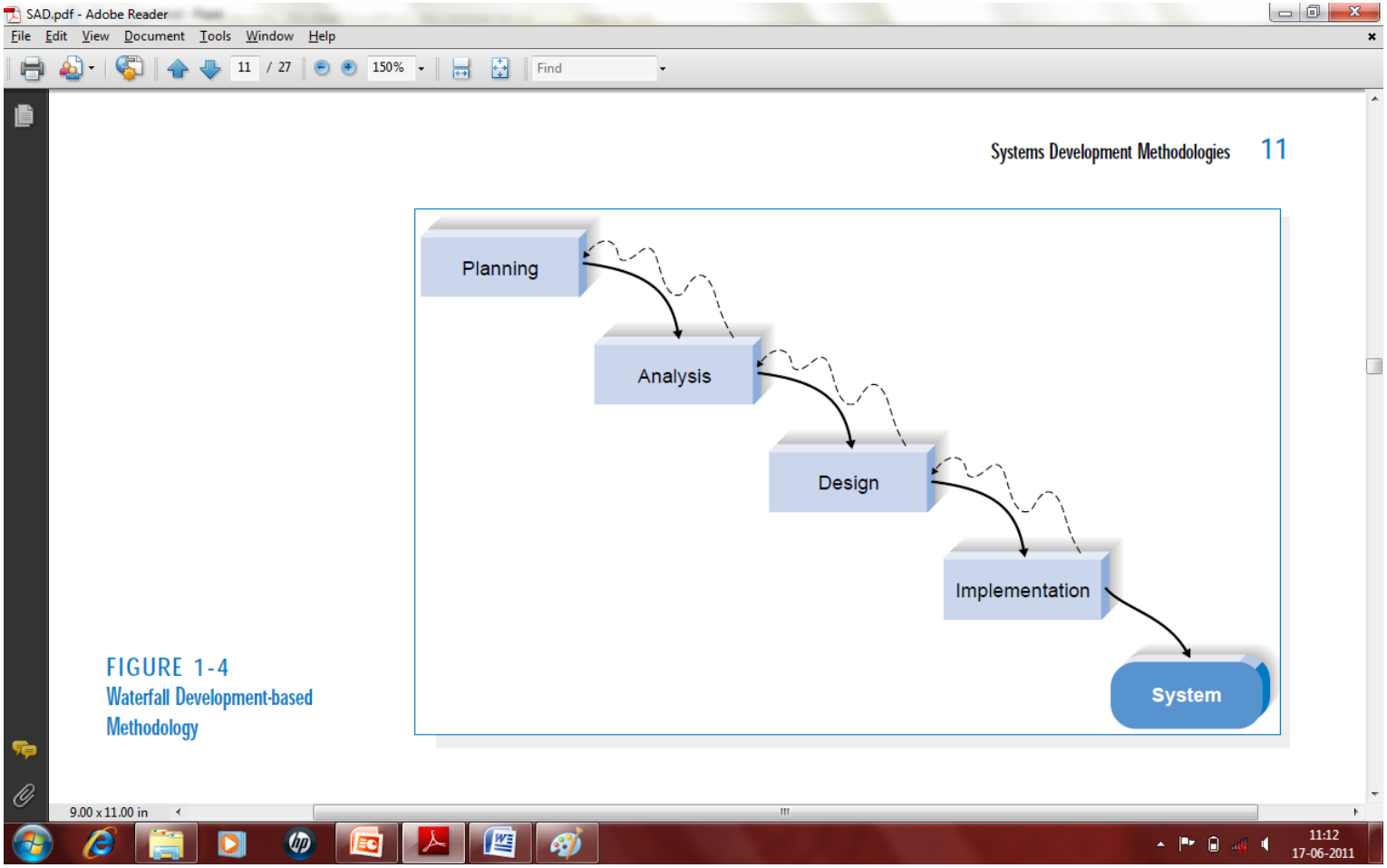
# Software Life Cycle

- Requirement Analysis
- Design
- Development (Construction)
- Testing
  - Unit
  - System
  - Integration
  - User Acceptance
- Deployment
- Maintenance

# Software Life Cycle Models

- Structured Methodologies
  - Waterfall
  - Parallel
- RAD Methodologies
  - Phased
  - Proto type
- Agile Methodology
  - Xtreme Programming
- Spiral

# Waterfall Model



# Waterfall Model

- In the olden days, Waterfall model was used to develop enterprise applications like
  - Customer Relationship Management (CRM) systems,
  - Human Resource Management Systems (HRMS),
  - Supply Chain Management Systems,
  - Inventory Management Systems,
  - Point of Sales (POS) systems for Retail chains etc.

# Current Scenario

- These days most project follow Agile Methodology, some form of Iterative model or one of the other models depending on their project specific requirement.

# Reasons why systems are obsolete by the time they were completed

- By the time the applications were developed in C, C++ etc, new languages (relatively speaking) like Java, .Net etc would replace them with web based functionality.

- Even if the application was developed using a new technology, factors like more competitors entering the market, cheaper alternatives becoming available, better functionality using newer technologies, change in customers requirement etc. increase the risk of developing an application over several years.

# When to use waterfall model

- This model is used only when the requirements are very well known, clear and fixed.
- Product definition is stable.
- Technology is understood.
- There are no ambiguous requirements
- Ample resources with required expertise are available freely
- The project is short.

# Some areas where Waterfall model was continued to be preferred

- Consider a system where human life is on the line, where a system failure could result in one or more deaths.
- In some countries, such mishaps could lead to imprisonment for those who are accountable.
- Consider a system where time and money were secondary considerations and human safety was first.

# Situations where Waterfall model was the preferred approach

- Development of Department Of Defense (DOD), military and aircraft programs followed Waterfall model in many organizations.
- This is because of the strict standards and requirements that have to be followed.
- In such industries, the requirements are known well in advance and contracts are very specific about the deliverable of the project.
- DOD Agencies typically considered Waterfall model to be compatible with their acquisition process and rigorous oversight process required by the government.

# Waterfall model was also used in

- banking,
- healthcare,
- control system for nuclear facilities,
- space shuttles etc

# Advantages of Waterfall model

- This model is simple and easy to understand and use.
- It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- In this model phases are processed and completed one at a time. Phases do not overlap.
- Waterfall model works well for smaller projects where requirements are clearly defined and very well understood.

# Disadvantages of Waterfall model

- Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing.

# Parallel Development Model

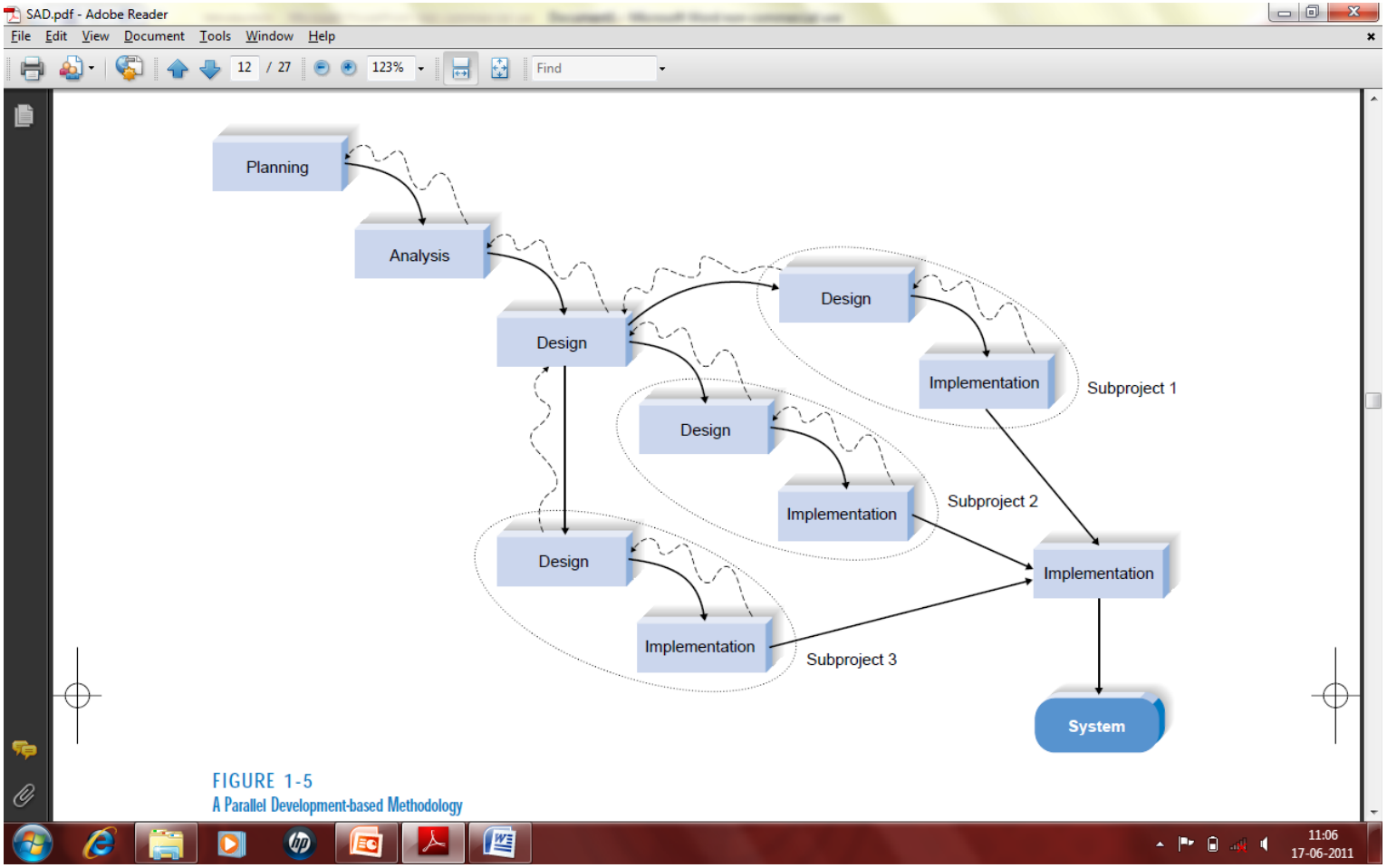
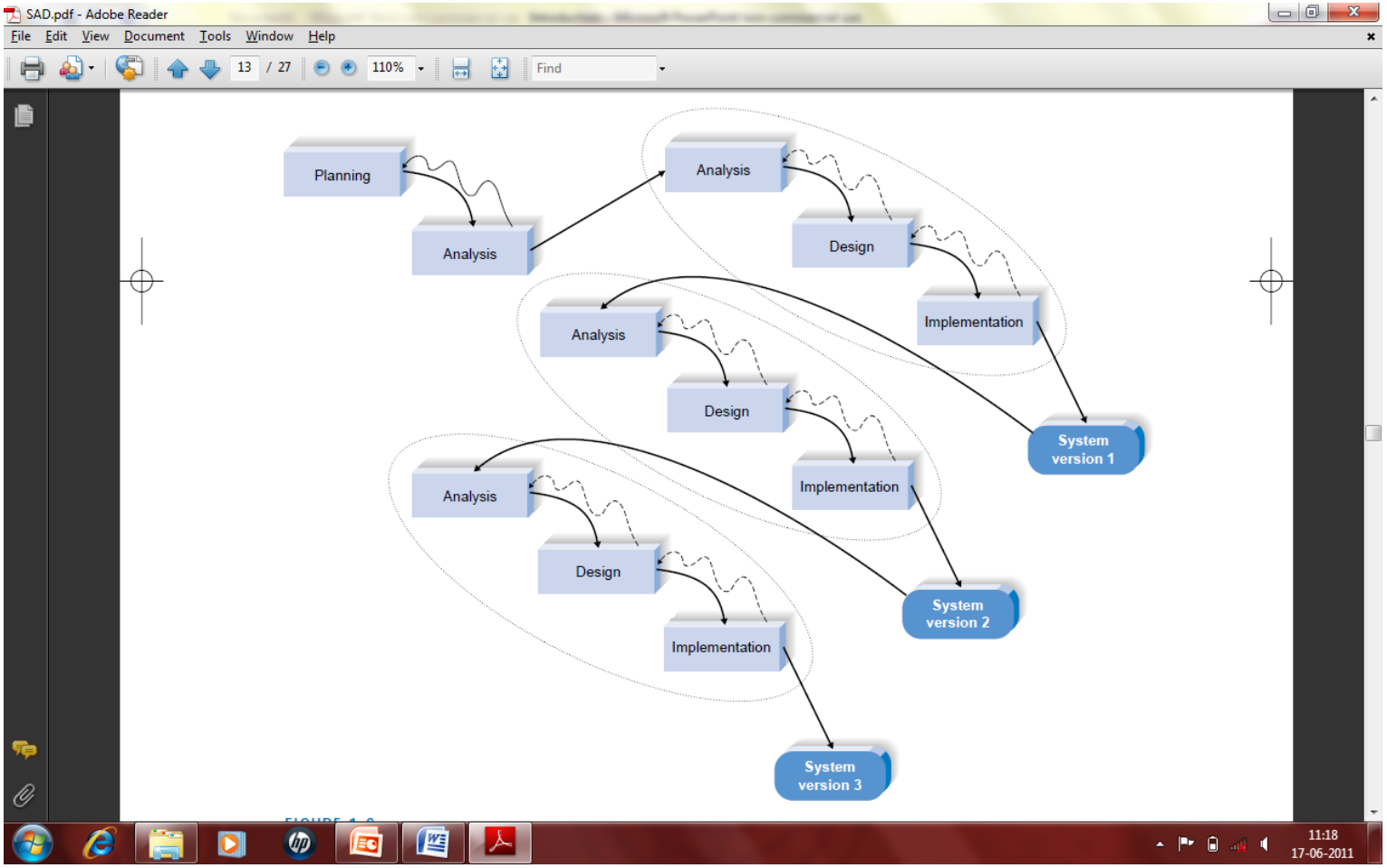



FIGURE 1-5
A Parallel Development-based Methodology

# Parallel Development Model

- Parallel Software Development simply means that instead of following a strictly linear path for a project where coding is done chronologically, various parts of the project are simultaneously worked on.
- So,  at any time, various versions of the project exist, having different features and proceeding along different directions.
- They may merge again to form the final project.
- This approach saves time and quickens the process while requiring more resources.

# Phased approach Model



# Rapid Application Development

- It is a type of incremental model.
- In RAD model the components or functions are developed in parallel as if they were mini projects.
- The developments are time boxed, delivered and then assembled into a working prototype.

# Time Boxing

- Timeboxing is allotting a fixed, maximum unit of time for an activity.
- That unit of time is called a time box.
- The goal of timeboxing is to define and limit the amount of time dedicated to an activity.
- Timeboxing also encourages teams to start getting work done immediately.

# RAD

- This can quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements.

# RAD

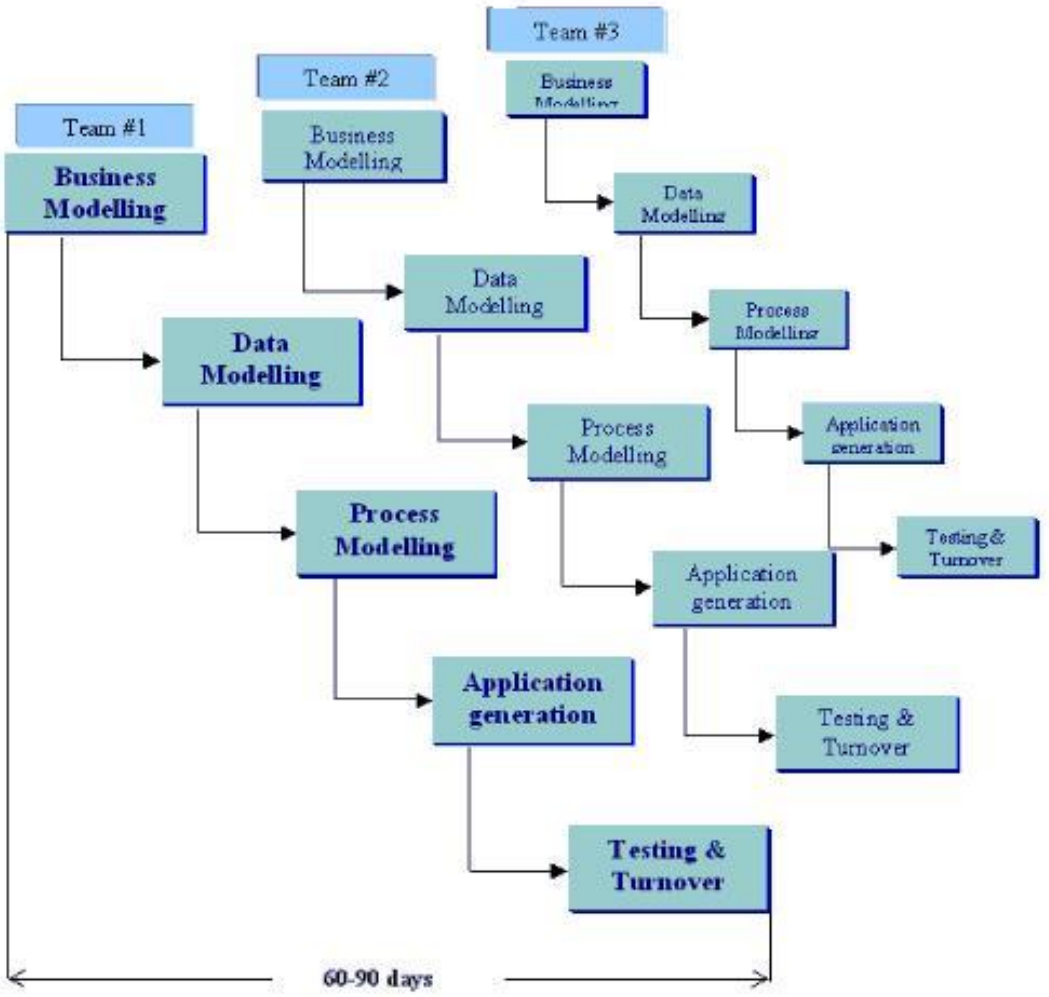



Figure 1.5 – RAD Model

# Phases in RAD

- **Business modeling:** The information flow is identified between various business functions.
- **Data modeling:** Information gathered from business modeling is used to define data objects that are needed for the business.
- **Process modeling:** Data objects defined in data modeling are converted to achieve the business information flow to achieve some specific business objective. Description are identified and created for CRUD of data objects.
- **Application generation:** Automated tools are used to convert process models into code and the actual system.
- **Testing and turnover:** Test new components and all the interfaces.

# CRUD

- In computer programming, **create, read, update, and delete** (**CRUD**) are the four basic functions of persistent storage.
- Alternate words are sometimes used when defining the four basic functions of *CRUD*, such as *retrieve* instead of *read*, *modify* instead of *update*, or *destroy* instead of *delete*.

# Advantages of RAD model

- Reduced development time.
- Increases reusability of components
- Quick initial reviews occur
- Encourages customer feedback
- Integration from very beginning solves a lot of integration issues.

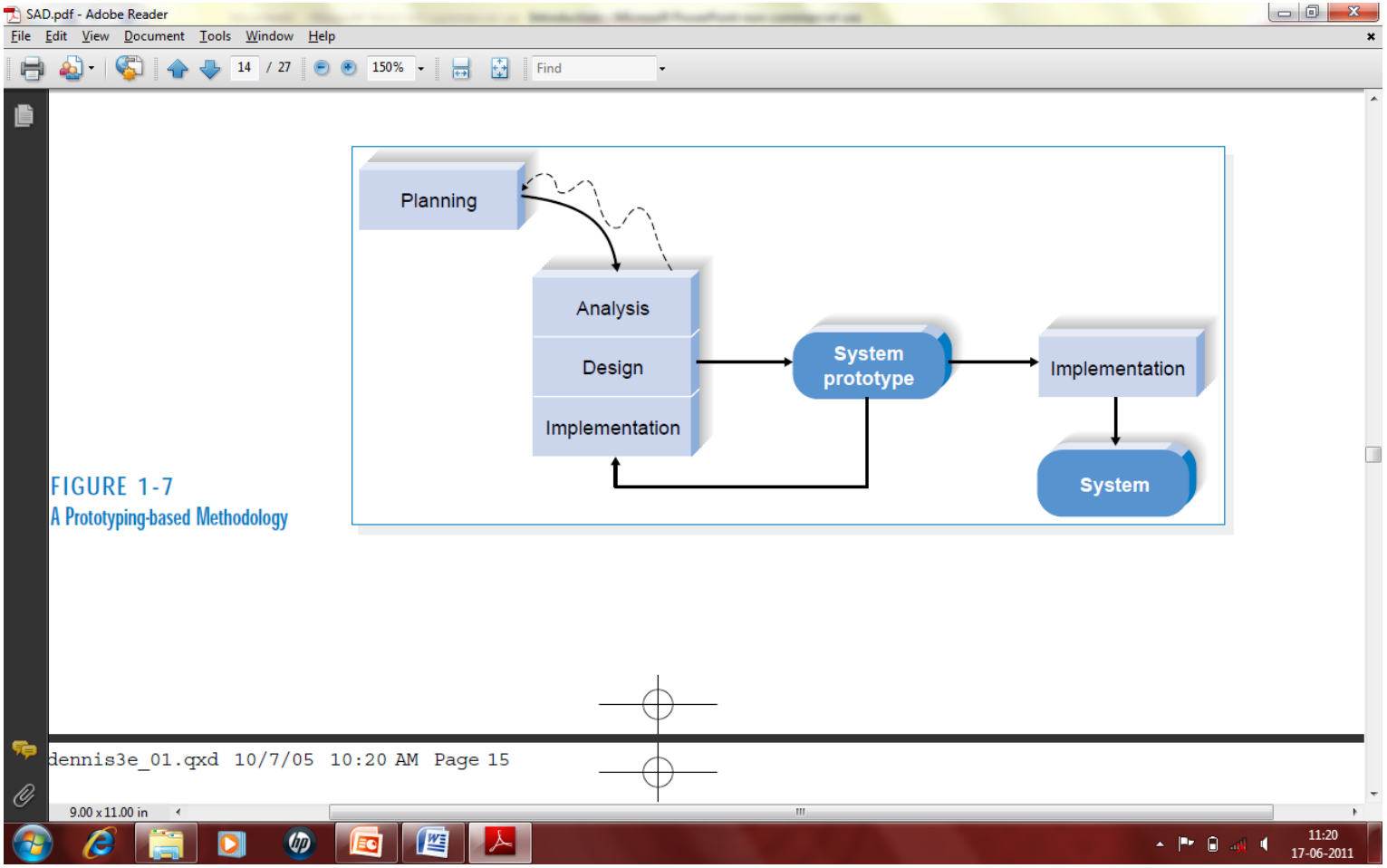
# Disadvantages of RAD model

- Depends on strong team and individual performances for identifying business requirements.
- Only systems that can be modularized can be built using RAD
- Requires highly skilled developers/designers.
- High dependency on modeling skills
- Inapplicable to cheaper projects as cost of modeling and automated code generation is very high.

# When to use RAD model

- RAD should be used when there is a need to create a system that can be modularized in 2-3 months of time.
- It should be used if there's high availability of designers for modeling and the budget is high enough to afford their cost along with the cost of automated code generating tools.
- RAD model should be chosen only if resources with high business knowledge are available and there is a need to produce the system in a short span of time (2-3 months).

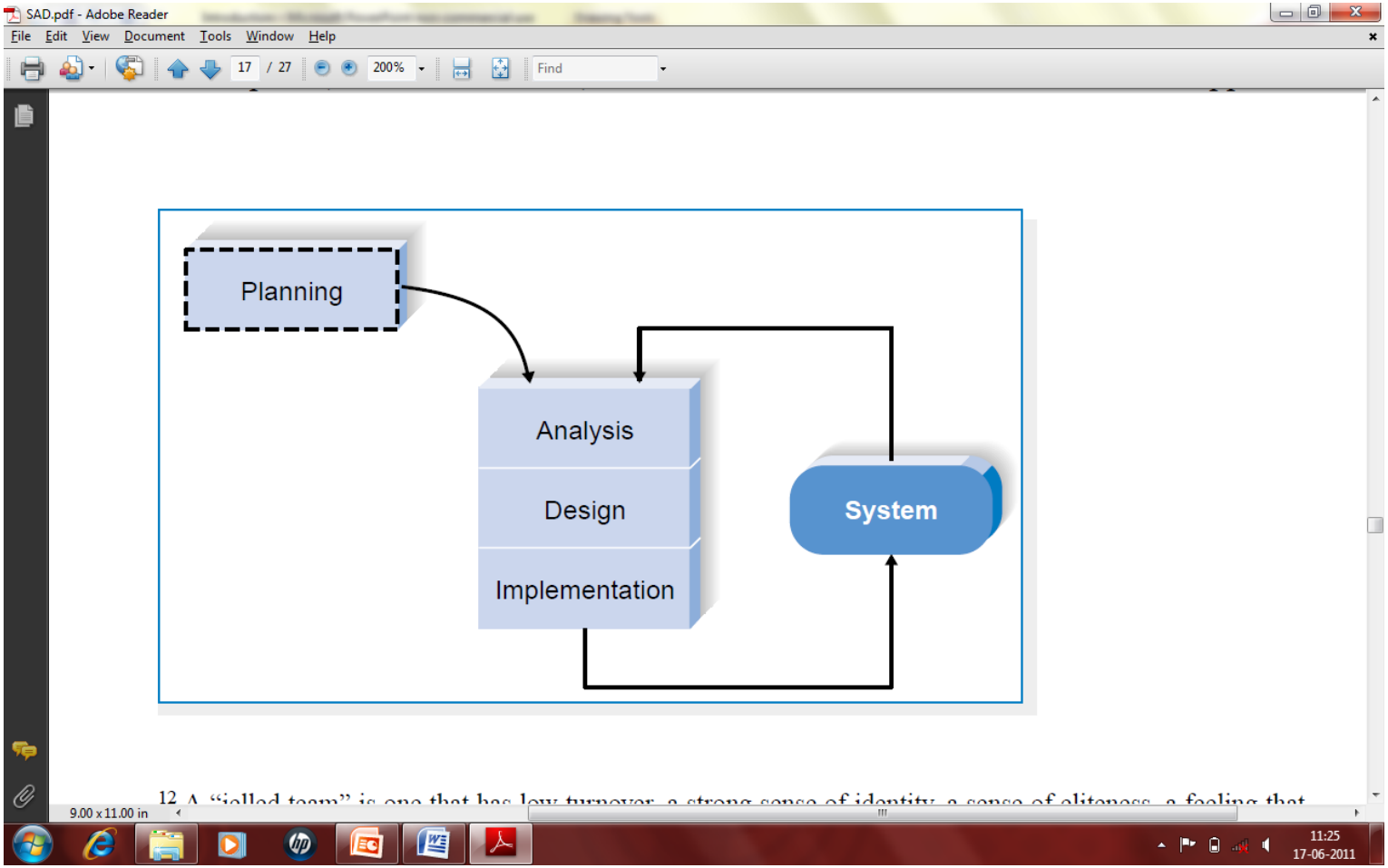
# Proto type based



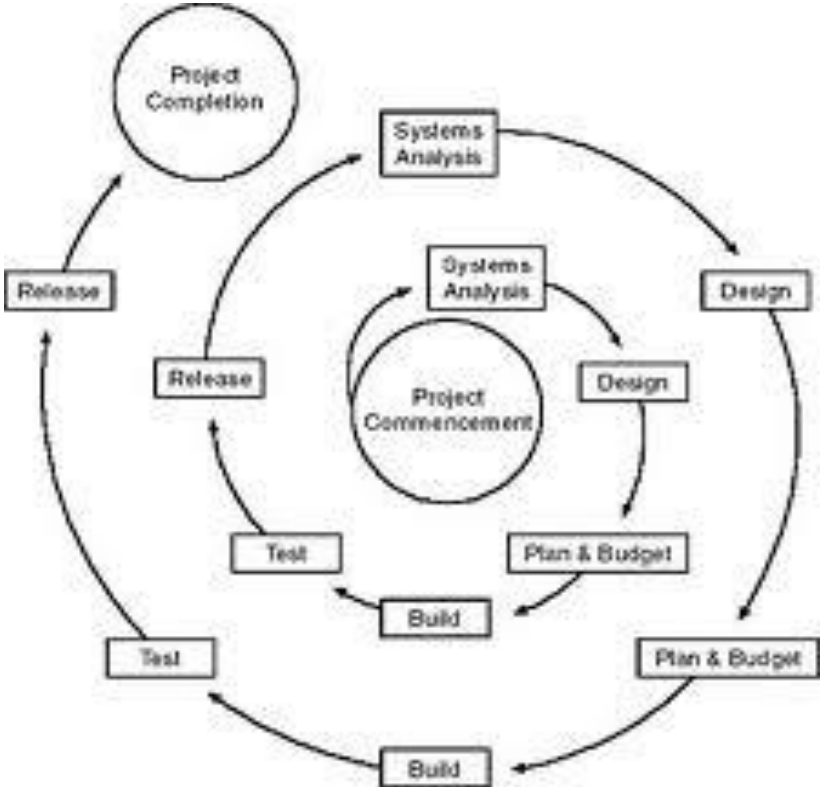
# Prototype model

- The basic idea in **Prototype model** is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements.
- This prototype is developed based on the currently known requirements.
- By using this prototype, the client can get an “actual feel” of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system.
- Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements.

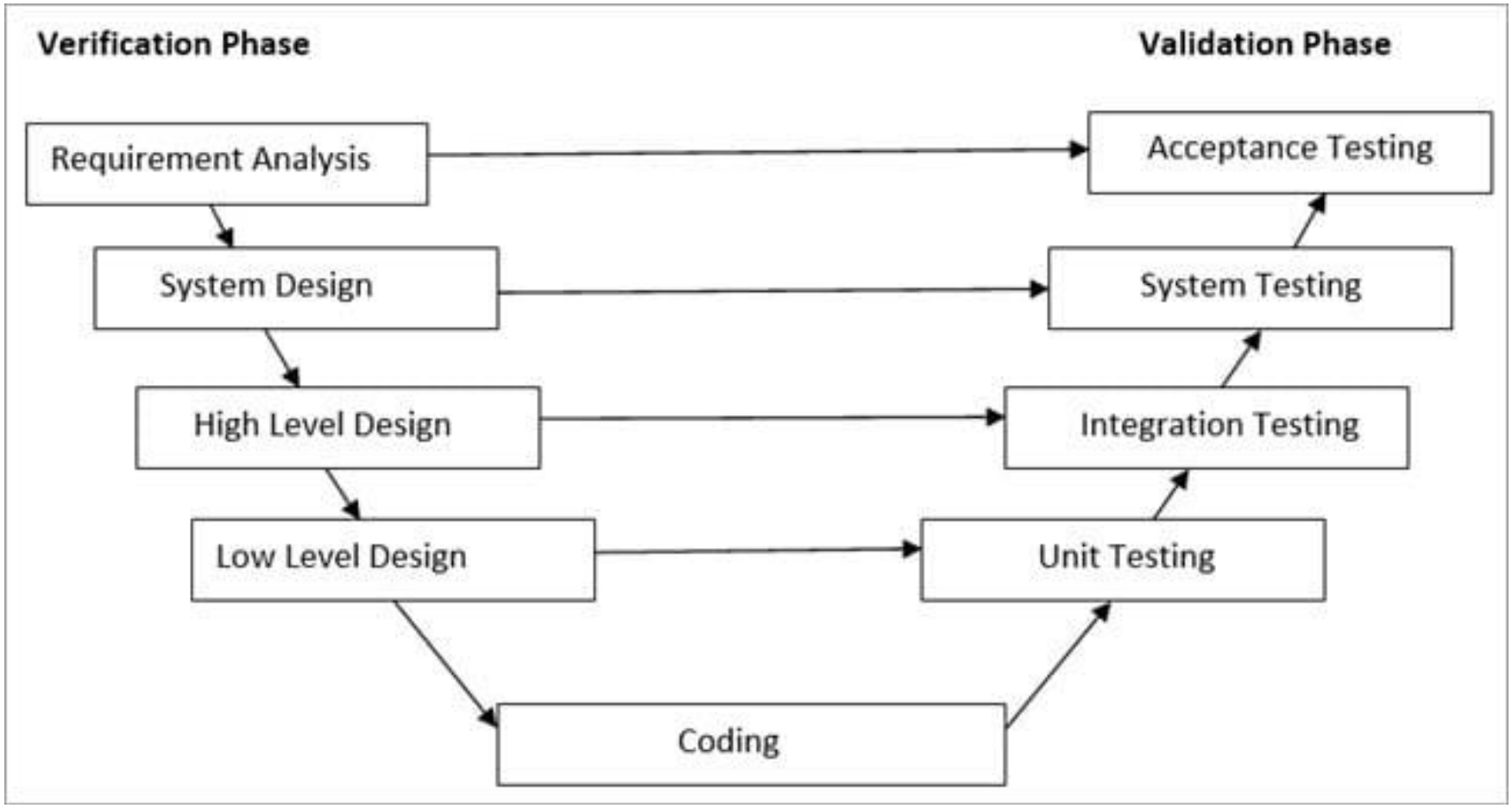
# Agile Development



# Spiral Model



# V shaped model



# Comparison

SAD.pdf - Adobe Reader

File Edit View Document Tools Window Help

18 / 27 150% Find

| Ability to Develop Systems | Structured Methodologies | | RAD Methodologies | | | Agile Methodologies |
|---|---|---|---|---|---|---|
| | Waterfall | Parallel | Phased | Prototyping | Throwaway Prototyping | XP |
| with Unclear User Requirements | Poor | Poor | Good | Excellent | Excellent | Excellent |
| with Unfamiliar Technology | Poor | Poor | Good | Poor | Excellent | Poor |
| that are Complex | Good | Good | Good | Poor | Excellent | Poor |
| that are Reliable | Good | Good | Good | Poor | Excellent | Good |
| with a Short Time Schedule | Poor | Good | Excellent | Excellent | Good | Excellent |
| with Schedule Visibility | Poor | Poor | Excellent | Excellent | Good | Good |

FIGURE 1-10
Criteria for Selecting a Methodology

[14] Many of the observations described on the utility of XP as a development approach were based on conversations with Brian Henderson-Sellers.

9.00 x 11.00 in

5:37 PM
11/6/2012