Chapter 3: Agile

- **Rapid** development and delivery is now often the most important requirement for software systems

- Specification, design and implementation are **inter-leaved**

- System is developed as a series of versions with stakeholders involved in version evaluation

- User interfaces are often developed using an IDE and graphical toolset

***Working software > comprehensive documentation***

***Customer collaboration > contract negotiation***

***Responding to change > following a plan***

-**Customers** should be closely involved throughout the development process. Their role is provide **and prioritize new system requirements** and to **evaluate the iterations** of the system

-The skills of the development team should be recognized and exploited. Team members should be left to develop their own ways of working without prescriptive processes **(no specific rules to follow)**

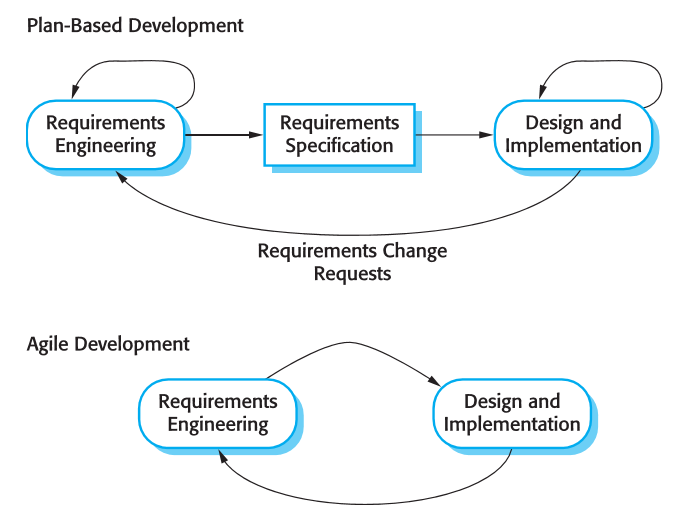
**-** **Expect the system requirements to change** and so design the system to accommodate these changes

**-** Wherever possible,actively work to **eliminate complexity**

**-Applicability**:

- Small or medium-sized products for sale

- Cases where customer displays commitment to become involved in the process (won’t get bored of iterations)

**-Problems** arise when trying to scale for larger systems

**-Issues:**

- Difficulty keeping the customer **interested**

- Team members may be **unsuited** for intensity of involvement

- Hard to prioritize changes with **multiple** customers

- Difficult for large companies with aged work cultures to switch to an informal system that gives power to developers

- Companies spend more on maintenance than they do on new software => Agile-made software must support such maintenance

**-Balance:**

Most projects include elements of **plan-driven** and **agile** processes. This balance depends on:

|  |  |  |
| --- | --- | --- |
|  | **Plan-Driven** | **Agile** |
| **Documentation** | Detailed specification and design before starting | Details are only set for the next incrimination |
| **Size** | Large systems | Small-to-medium co-located team |
| **Analysis** | Analysis before implementation required | No prior analysis |
| **Lifetime** | Long Lifetime (due to documentation) | Standard Lifetime |
| **Tools** | Standard Tools | Advanced tools for tracking evolution design |
| **Team** | Can be distributed | Preferably co-located |
| **Skills** | Normal Developers | Requires more skilled developers |
| **Confirmation** | Regulator approval is needed (documentation) | No external confirmation needed |