Waterfall Model

Presents a systematic sequential approach to development

Advantages and Limitations

- Better than a haphazard approach
- Requires that requirements are well-specified and unchanging
- Assumes a sequential flow, which real projects rarely follow
- Working version becomes available very late (after coding)
- Developers may have to wait due to dependent tasks

Conclusion

- When the requirements are well understood, it is reasonable to use this approach
Prototyping Model

A prototype is developed to understand the requirements

Advantages and Limitations

- Misunderstandings, ambiguities and missing functions may be identified
- User sees the model at an early stage
- The user may ask for the finished product via “a few fixes” to the prototype
- Developers may make implementation compromises

Conclusion

- When requirements are not well understood, it may be reasonable to use this approach
- A prototype’s purpose is to understand requirements. A better (good quality) product must be subsequently developed.
Incremental Model

Delivers software in increments, each increment is a working product and adds to the functionality of the previous increment

Advantages and Limitations

- Business pressures may be met
- Technical risks may be managed
- Fewer resources may be used to proceed
- Client does not have to pay for entire software together
- Increments may be difficult to define
- Software may be difficult to maintain

Conclusion

- When the “core” product is well understood and increments can be easily defined, it is reasonable to use this approach
Spiral Model

Delivers the software in a series of incremental releases

Advantages and Limitations

- Risks are resolved before they become problematic
- Requires risk analysis expertise

Conclusion

- When risks are high and need to be resolved, it is reasonable to use this approach