

Different plan-driven software development processes

Inf2-SEPP

Adriana Sejfia

School of Informatics, University of Edinburgh

This lecture

Plan-driven processes:

- Reminder
- The Waterfall Model
- The Spiral Model

Plan-driven processes- reminder

“Plan-driven processes are processes where all the process activities are planned in advance and progress is measured against this plan” (Ian Sommerville)

The waterfall model

- Introduced by Winston W. Royce in 1970

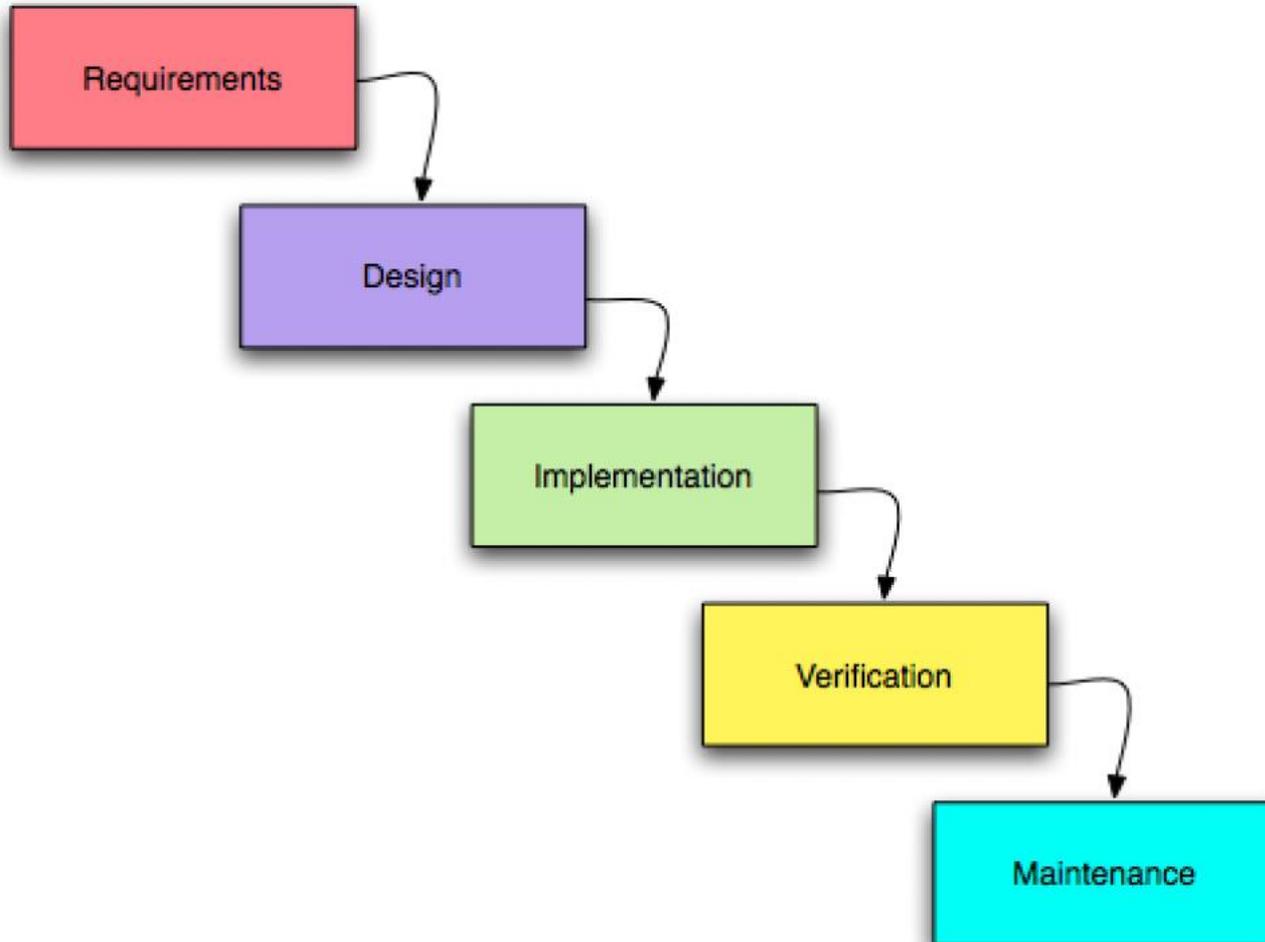


Image from Wikipedia

The waterfall model: pros and cons

Pros:

- better than no process at all – makes clear that requirements engineering, verification etc. important.

The waterfall model: pros and cons

Pros:

- better than no process at all – makes clear that requirements engineering, verification etc. important.
- time spent early can reduce higher costs later.

The waterfall model: pros and cons

Pros:

- better than no process at all – makes clear that requirements engineering, verification etc. important.
- time spent early can reduce higher costs later.
- emphasis on documentation ensures that knowledge is not lost when team members leave.

The waterfall model: pros and cons

Pros:

- better than no process at all – makes clear that requirements engineering, verification etc. important.
- time spent early can reduce higher costs later.
- emphasis on documentation ensures that knowledge is not lost when team members leave.
- easily understandable, explainable, to manage.

The waterfall model: pros and cons

Pros:

- better than no process at all – makes clear that requirements engineering, verification etc. important.
- time spent early can reduce higher costs later.
- emphasis on documentation ensures that knowledge is not lost when team members leave.
- easily understandable, explainable, to manage.

Cons:

- inflexible and unrealistic in many domains in practice: e.g., verification could show up problems with requirements.

The waterfall model: pros and cons

Pros:

- better than no process at all – makes clear that requirements engineering, verification etc. important.
- time spent early can reduce higher costs later.
- emphasis on documentation ensures that knowledge is not lost when team members leave.
- easily understandable, explainable, to manage.

Cons:

- inflexible and unrealistic in many domains in practice: e.g., verification could show up problems with requirements.
- slow and expensive – in an attempt to avoid problems later, end up “gold plating” early phases, e.g., designing something elaborate enough to support the requirements you suspect you’ve missed, so that functionality for them can be added in coding.

The waterfall model: pros and cons

Pros:

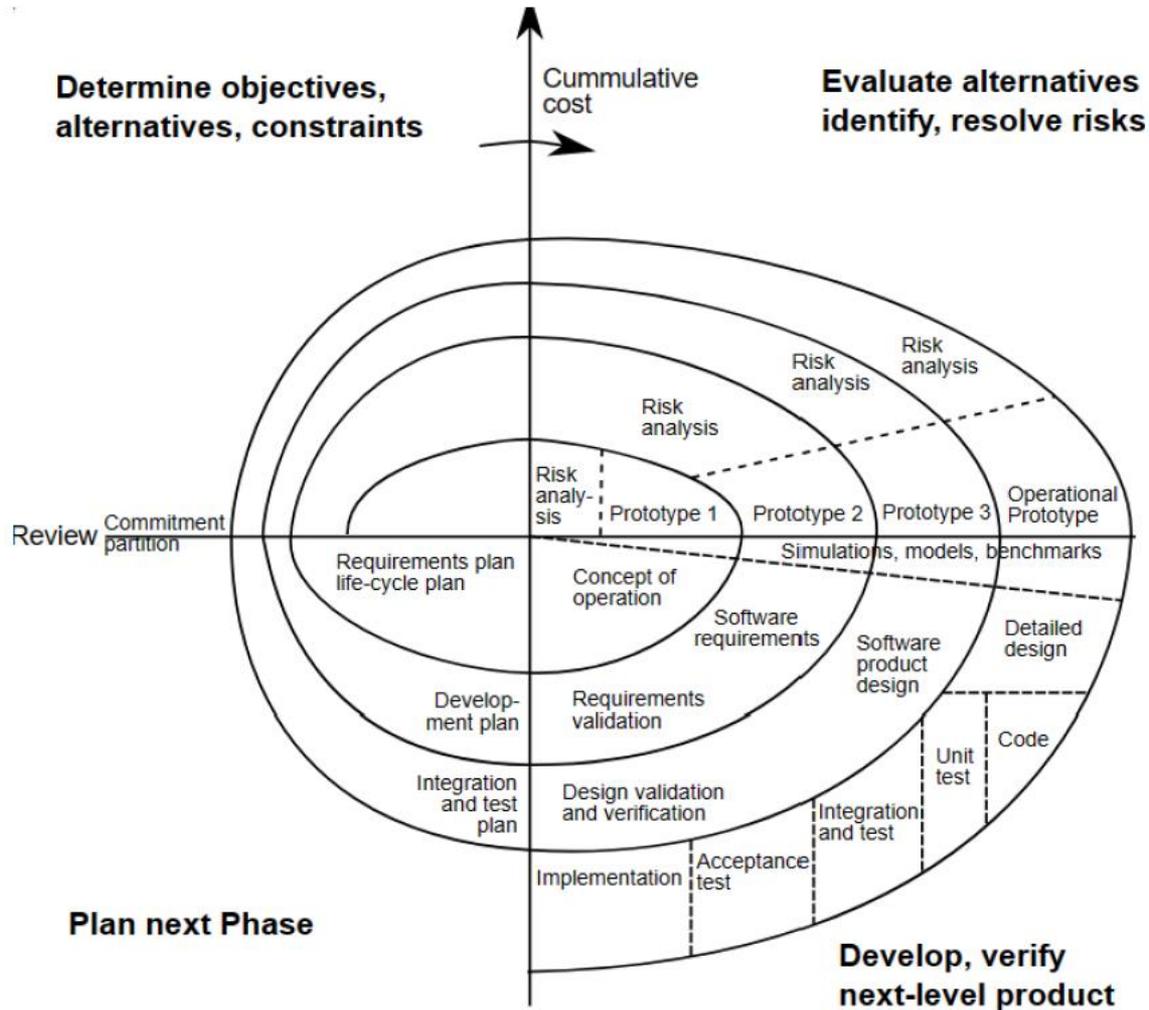
- better than no process at all – makes clear that requirements engineering, verification etc. important.
- time spent early can reduce higher costs later.
- emphasis on documentation ensures that knowledge is not lost when team members leave.
- easily understandable, explainable, to manage.

Cons:

- inflexible and unrealistic in many domains in practice: e.g., verification could show up problems with requirements.
- slow and expensive – in an attempt to avoid problems later, end up “gold plating” early phases, e.g., designing something elaborate enough to support the requirements you suspect you’ve missed, so that functionality for them can be added in coding.
- high amounts of risk and uncertainty

The spiral model

- Barry Boehm. 1986.



The spiral model

Phases of a single cycle:

1. **Determine objectives, alternative means of addressing them, constraints:** involving key stakeholders

The spiral model

Phases of a single cycle:

1. **Determine objectives, alternative means of addressing them, constraints:** involving key stakeholders
2. **Evaluate alternatives, identify, resolve risks:** results in identification of cost-effective strategy to resolve the risks which may involve risk-resolution techniques like prototyping, simulation, benchmarking.

The spiral model

Phases of a single cycle:

1. **Determine objectives, alternative means of addressing them, constraints:** involving key stakeholders
2. **Evaluate alternatives, identify, resolve risks:** results in identification of cost-effective strategy to resolve the risks which may involve risk-resolution techniques like prototyping, simulation, benchmarking.
3. **Develop, verify next-level product:** decide on approach based on remaining risk; Can incorporate other processes like Waterfall, which is why Spiral is also called a process model generator. It requires good management.

The spiral model

Phases of a single cycle:

1. **Determine objectives, alternative means of addressing them, constraints:** involving key stakeholders
2. **Evaluate alternatives, identify, resolve risks:** results in identification of cost-effective strategy to resolve the risks which may involve risk-resolution techniques like prototyping, simulation, benchmarking.
3. **Develop, verify next-level product:** decide on approach based on remaining risk; Can incorporate other processes like Waterfall, which is why Spiral is also called a process model generator. It requires good management.
4. **Plan next phase:** Work reviewed, plans decided for next cycle.

The spiral model

Phases of a single cycle:

1. **Determine objectives, alternative means of addressing them, constraints:** involving key stakeholders
2. **Evaluate alternatives, identify, resolve risks:** results in identification of cost-effective strategy to resolve the risks which may involve risk-resolution techniques like prototyping, simulation, benchmarking.
3. **Develop, verify next-level product:** decide on approach based on remaining risk; Can incorporate other processes like Waterfall, which is why Spiral is also called a process model generator. It requires good management.
4. **Plan next phase:** Work reviewed, plans decided for next cycle.

-The cycle ends with key stakeholders reviewing outcomes and plans to ensure commitment for the next cycle.

The spiral model

Phases of a single cycle:

1. **Determine objectives, alternative means of addressing them, constraints:** involving key stakeholders
2. **Evaluate alternatives, identify, resolve risks:** results in identification of cost-effective strategy to resolve the risks which may involve risk-resolution techniques like prototyping, simulation, benchmarking.
3. **Develop, verify next-level product:** decide on approach based on remaining risk; Can incorporate other processes like Waterfall, which is why Spiral is also called a process model generator. It requires good management.
4. **Plan next phase:** Work reviewed, plans decided for next cycle.

-The cycle ends with key stakeholders reviewing outcomes and plans to ensure commitment for the next cycle.

-A key innovation is prominent role of risk.

The spiral model: pros and cons

Pros:

- risks considered and addressed early

The spiral model: pros and cons

Pros:

- risks considered and addressed early
- flexible, generic: objectives of next iteration informed by outputs of current iteration; what is developed in each iteration depends on the objectives and identified risks.

The spiral model: pros and cons

Pros:

- risks considered and addressed early
- flexible, generic: objectives of next iteration informed by outputs of current iteration; what is developed in each iteration depends on the objectives and identified risks.
- effort and level of detail informed by risk considerations

The spiral model: pros and cons

Pros:

- risks considered and addressed early
- flexible, generic: objectives of next iteration informed by outputs of current iteration; what is developed in each iteration depends on the objectives and identified risks.
- effort and level of detail informed by risk considerations
- early involvement of key stakeholders in planning and evaluation

The spiral model: pros and cons

Pros:

- risks considered and addressed early
- flexible, generic: objectives of next iteration informed by outputs of current iteration; what is developed in each iteration depends on the objectives and identified risks.
- effort and level of detail informed by risk considerations
- early involvement of key stakeholders in planning and evaluation
- control over costs, resources and quality, due to mix of planning including risk consideration and evaluation per iteration

The spiral model: pros and cons

Pros:

- risks considered and addressed early
- flexible, generic: objectives of next iteration informed by outputs of current iteration; what is developed in each iteration depends on the objectives and identified risks.
- effort and level of detail informed by risk considerations
- early involvement of key stakeholders in planning and evaluation
- control over costs, resources and quality, due to mix of planning including risk consideration and evaluation per iteration

Cons:

- high management effort required to make decisions on what will be done in each iteration

The spiral model: pros and cons

Pros:

- risks considered and addressed early
- flexible, generic: objectives of next iteration informed by outputs of current iteration; what is developed in each iteration depends on the objectives and identified risks.
- effort and level of detail informed by risk considerations
- early involvement of key stakeholders in planning and evaluation
- control over costs, resources and quality, due to mix of planning including risk consideration and evaluation per iteration

Cons:

- high management effort required to make decisions on what will be done in each iteration
- know-how in risk analysis and risk management essential, but often not available

The spiral model: pros and cons

Pros:

- risks considered and addressed early
- flexible, generic: objectives of next iteration informed by outputs of current iteration; what is developed in each iteration depends on the objectives and identified risks.
- effort and level of detail informed by risk considerations
- early involvement of key stakeholders in planning and evaluation
- control over costs, resources and quality, due to mix of planning including risk consideration and evaluation per iteration

Cons:

- high management effort required to make decisions on what will be done in each iteration
- know-how in risk analysis and risk management essential, but often not available
- unsuitable for smaller systems with manageable risk

Example

A university IT department commissions a team of eight developers to build a new **student timetabling system**. Because the system must integrate with several legacy databases and meet strict regulatory requirements, the team begins by spending two months producing a **detailed requirements specification and system architecture document**.

The project manager insists that all requirements must be **approved and signed off by stakeholders** before development begins. The team also produces detailed design diagrams and a full project schedule for the entire 12-month project.

During development, programmers work on components according to the design specification. Changes to requirements are discouraged; when a department requests a modification, the request must go through a **formal change control board**, which evaluates its cost and impact before deciding whether it should be added to the project plan.

Testing is scheduled after the implementation phase. A separate testing group validates the system against the original requirements document before deployment.





Example

Suppose the team instead worked in **two-week iterations**, delivering partial timetable features and refining requirements with stakeholders each cycle. Which process style would this most closely resemble?



Reading

- Essential: Sommerville SE Chapter 2 only introduction and section 2.1
- Recommended: Browse the web to learn more about the:
 - Waterfall model
 - Spiral model (e.g. browse its phases and invariants from <https://ieeexplore.ieee.org/document/59>, https://insights.sei.cmu.edu/documents/5439/2000_003_001_13655.pdf)