

Process iteration

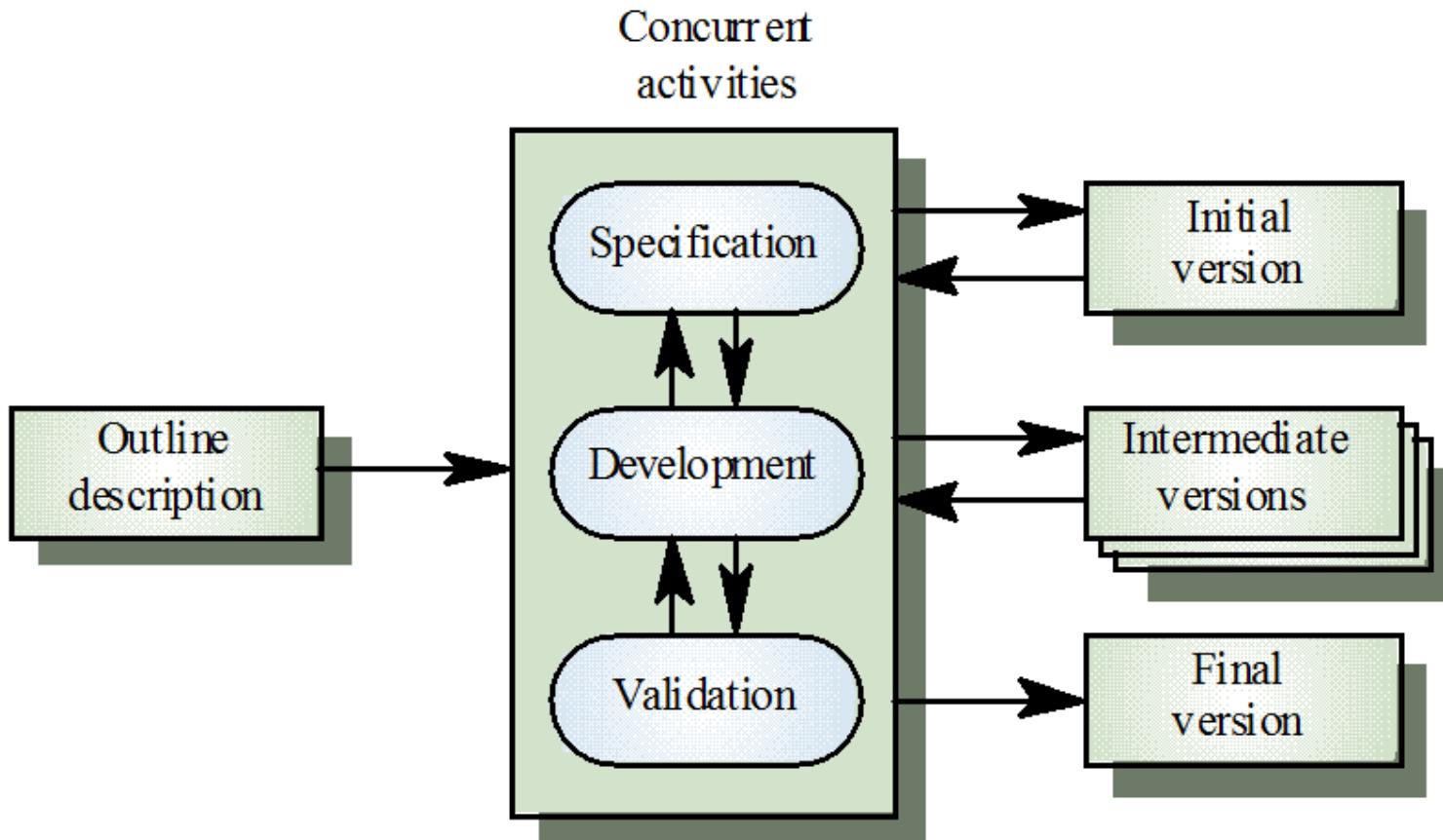
- Requirements ALWAYS evolve in the course of a project so process iteration where earlier stages are reworked is always part of the process for large products
- Iteration can be applied to any of the generic process models
- Two (related) approaches
 - Incremental development
 - Spiral development

Incremental development

- The product is developed and delivered in increments after establishing an overall architecture
- Requirements and specifications for each increment may be developed
- Users may experiment with delivered increments while others are being developed. Therefore, these serve as a form of prototype
- Intended to combine some of the advantages of prototyping but with a more manageable process and better structure

Modello incrementale

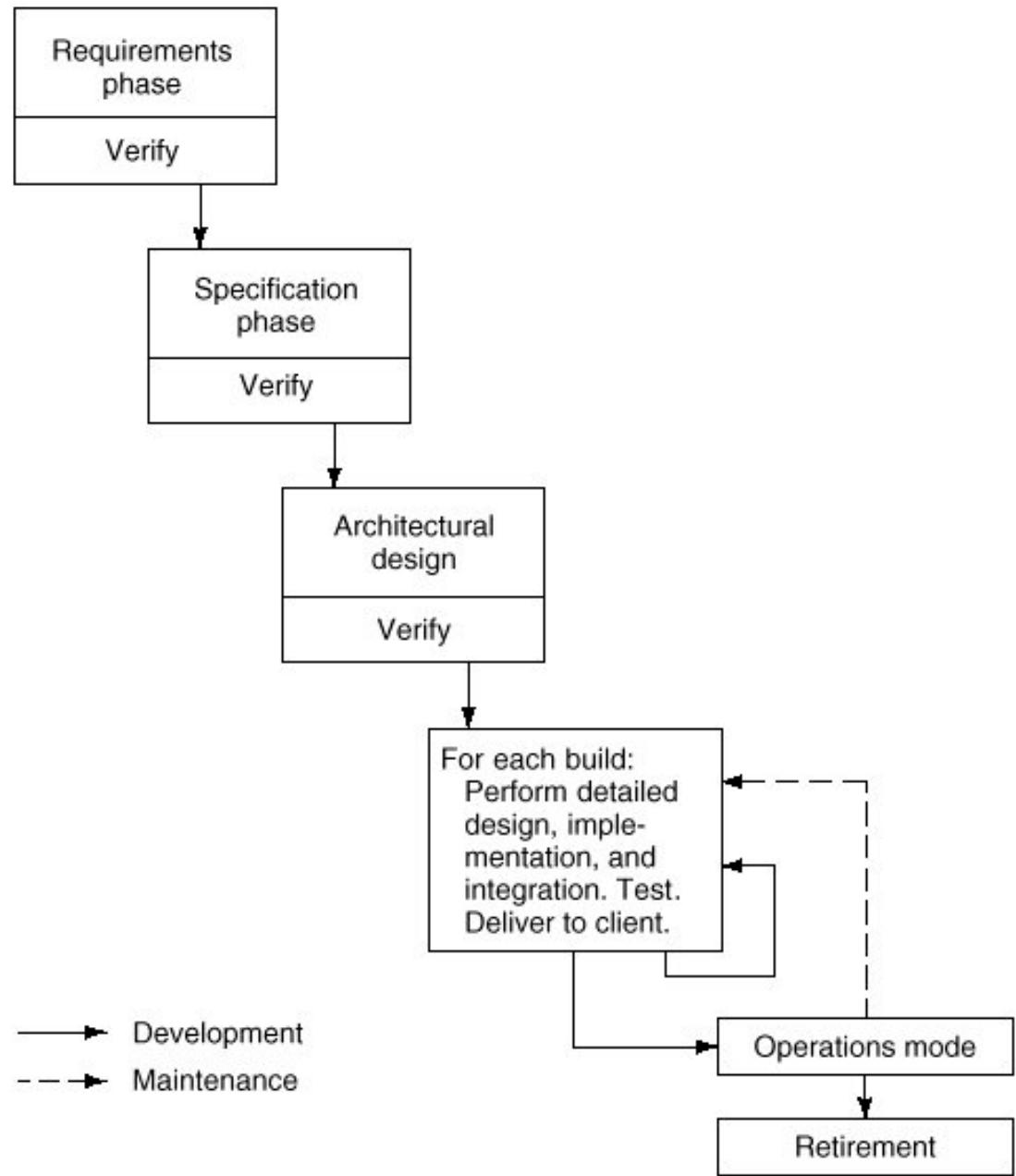
- Il prodotto software viene sviluppato e rilasciato per incrementi (*build*) successivi



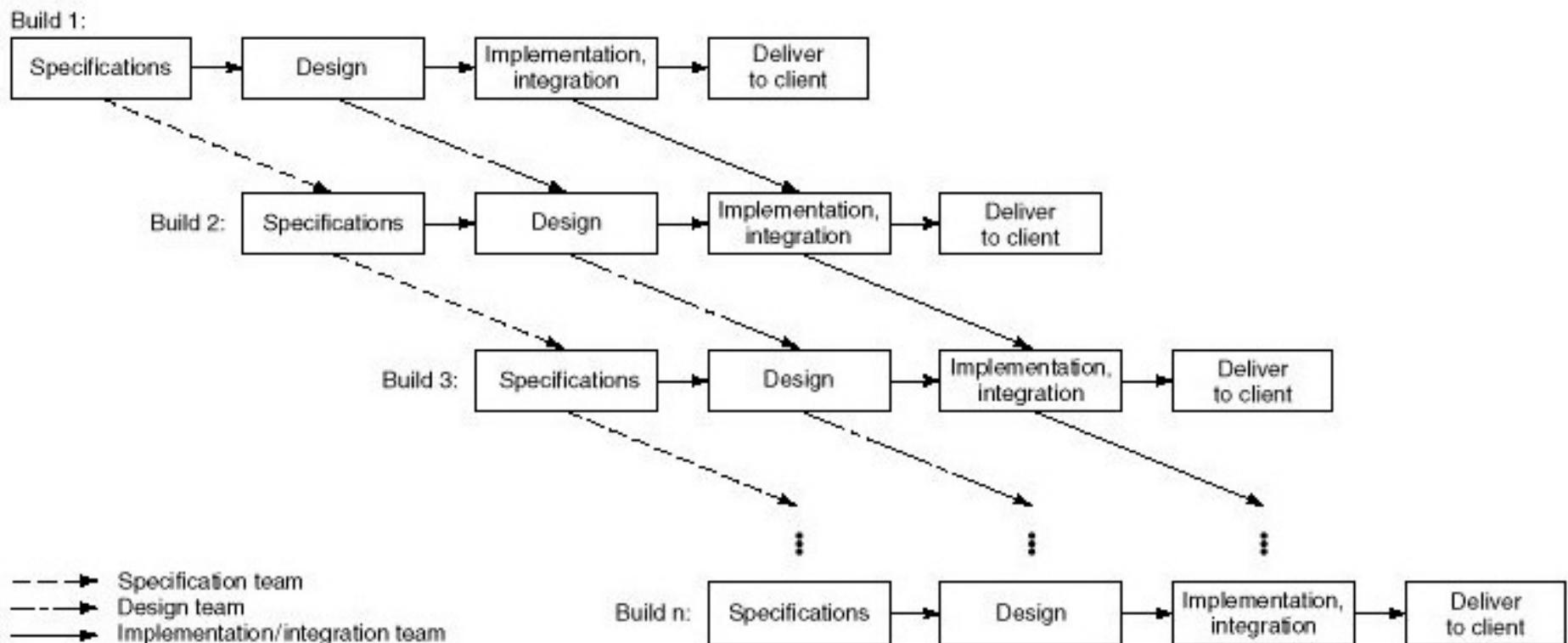
Modello incrementale (cont.)

- Include aspetti tipici del modello basato su *rapid prototyping* (l'utente può sperimentare l'utilizzo del prodotto contenente gli incrementi consegnati, mentre i restanti sono ancora in fase di sviluppo)
- Si rivela efficace quando il cliente vuole continuamente verificare i progressi nello sviluppo del prodotto e quando i requisiti subiscono modifiche
- Può essere realizzato in due versioni alternative:
 - versione con *overall architecture*
 - versione senza *overall architecture* (più rischiosa)

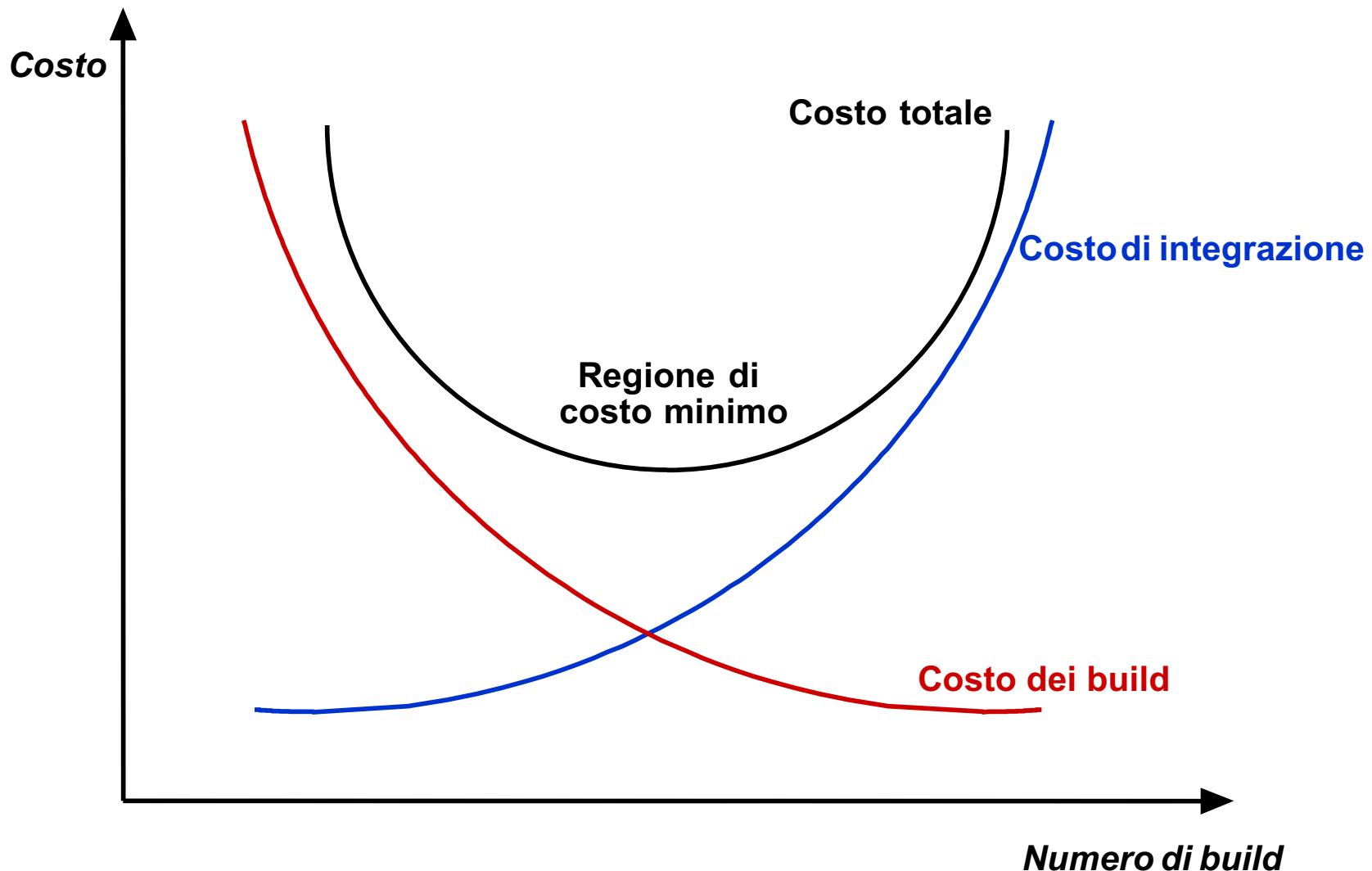
Versione con *overall* architecture



Versione senza *overall architecture*



Impatto sui costi del software



Confronto con modello a cascata

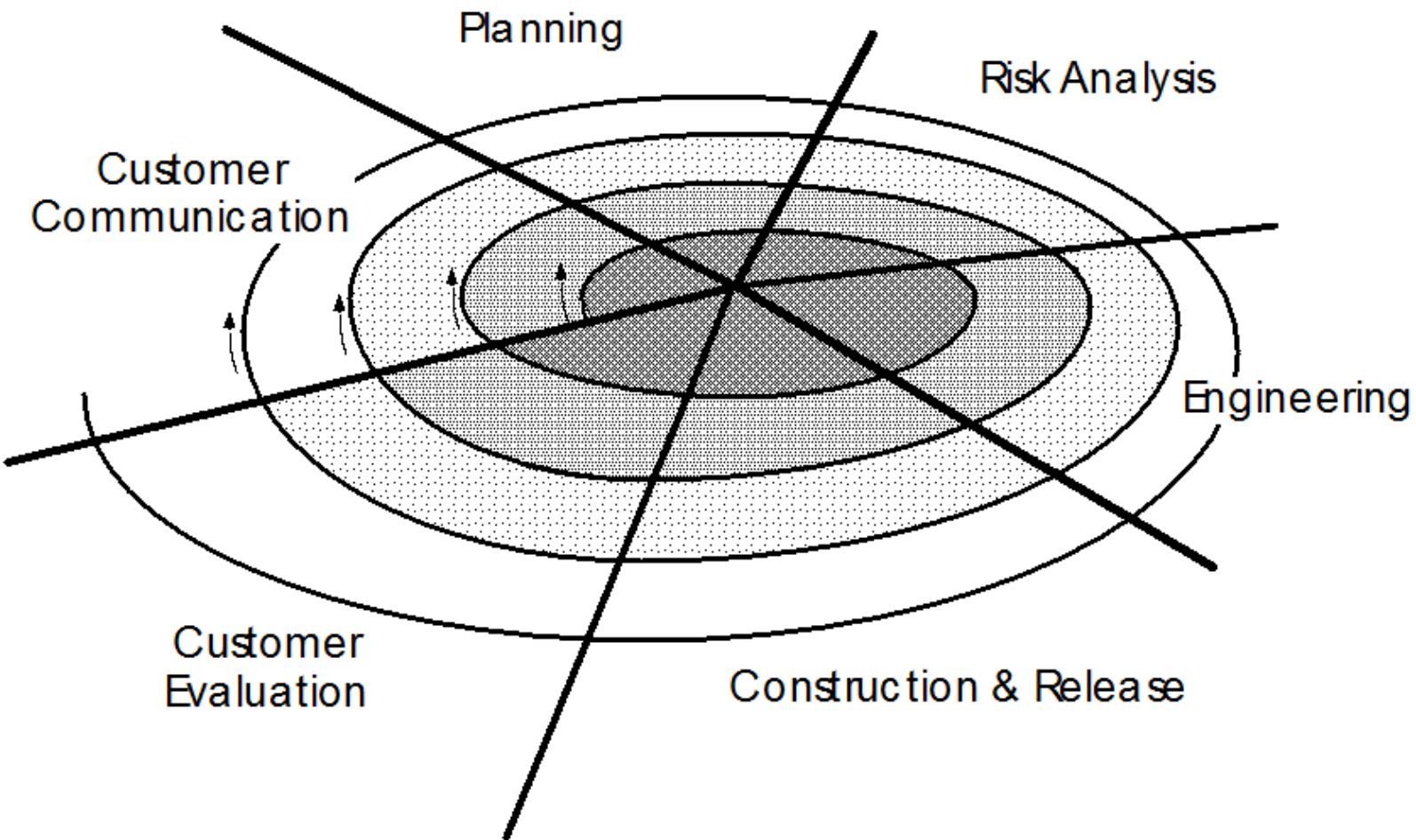
Modello a cascata

- Requisiti “congelati” al termine della fase di specifica
- Feedback del cliente solo una volta terminato lo sviluppo
- Fasi condotte in rigida sequenza (l’output di una costituisce input per la successiva)
- Prevede fasi di progetto dettagliato e codifica dell’intero prodotto
- Team di sviluppo costituito da un numero elevato di persone

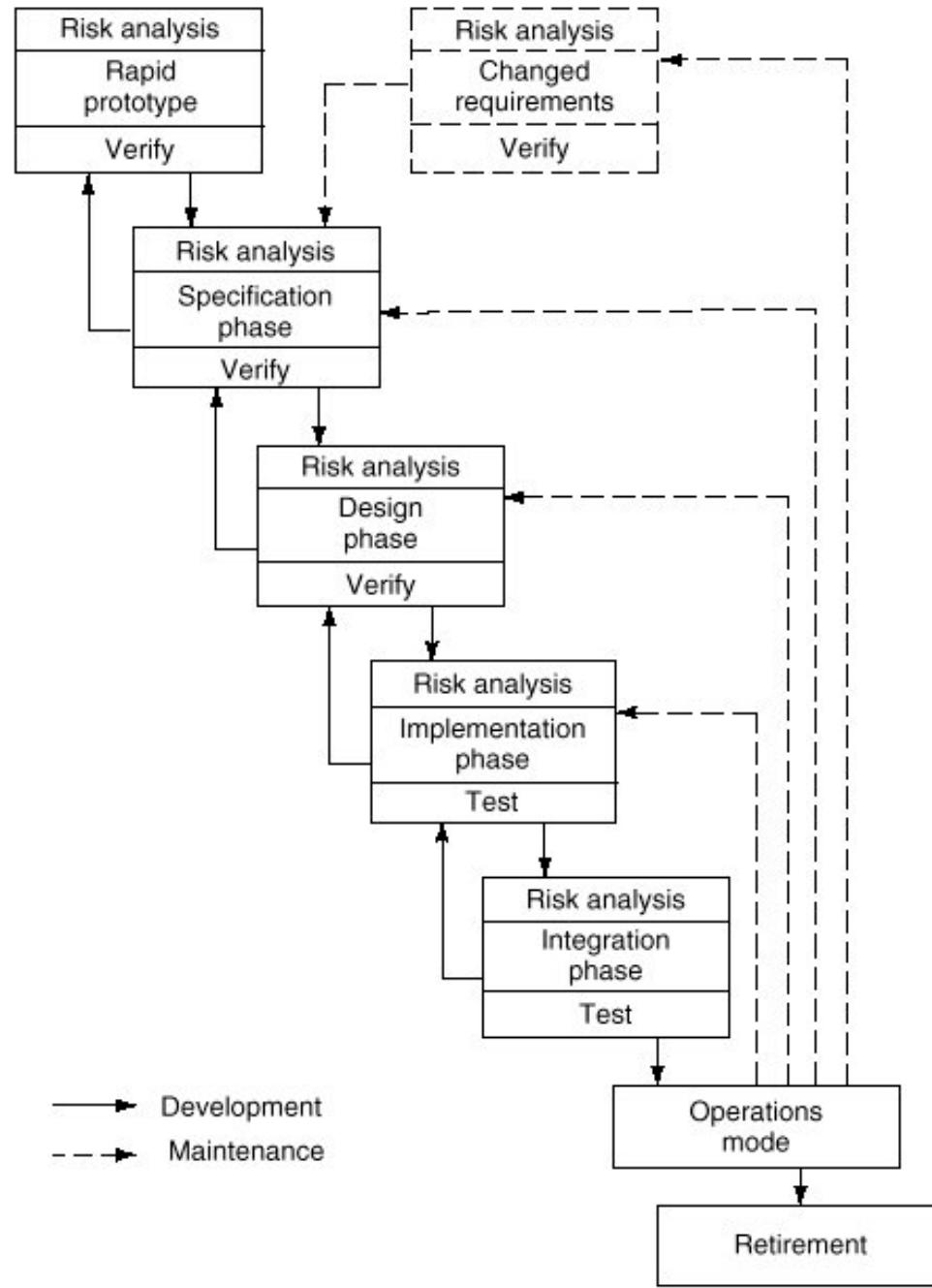
Modello incrementale

- Requisiti suddivisi in classi di priorità e facilmente modificabili
- Continuo feedback da parte del cliente durante lo sviluppo
- Fasi che possono essere condotte in parallelo
- Progetto dettagliato e codifica vengono effettuate sul singolo *build*
- Differenti team di sviluppo, ciascuno di piccole dimensioni

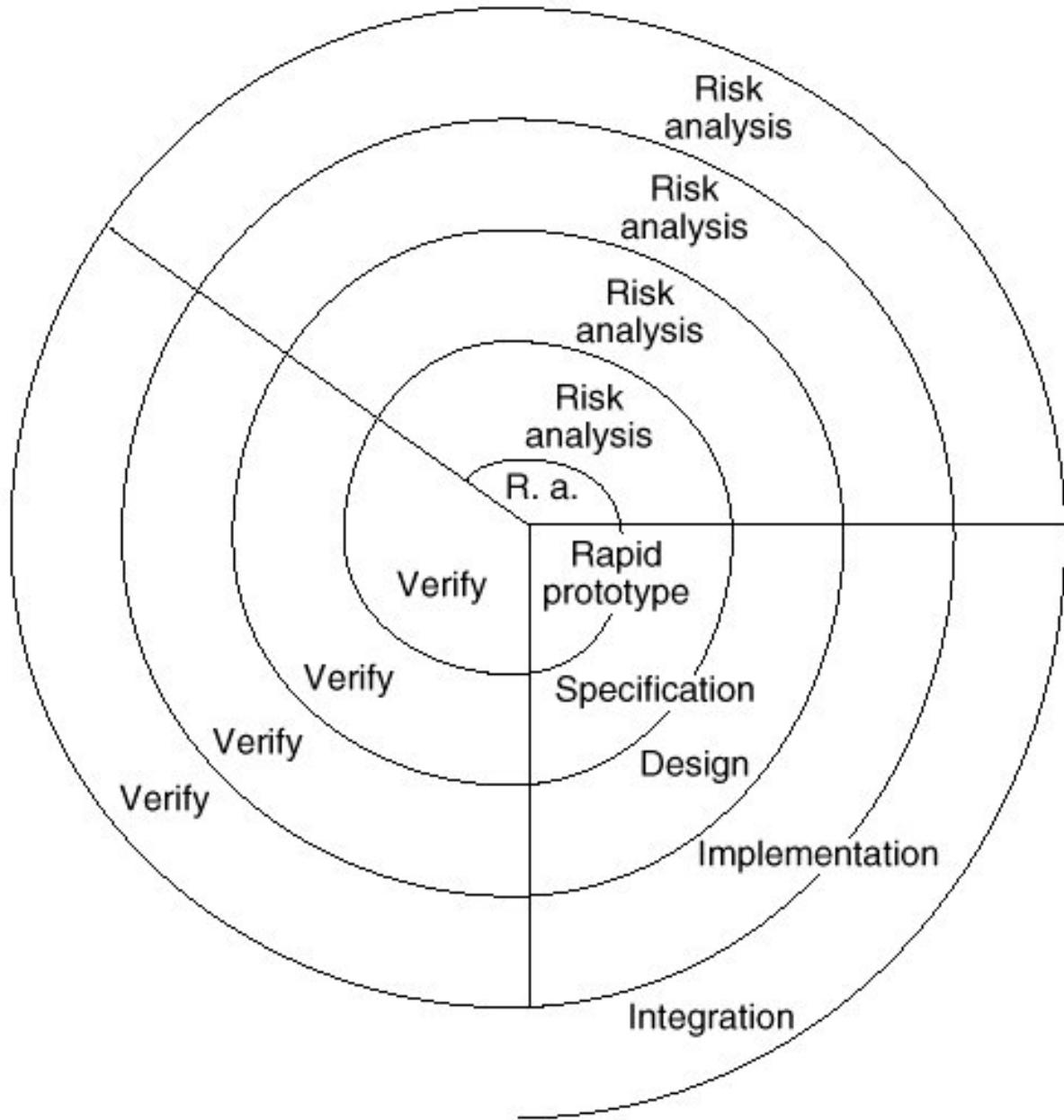
Modello a spirale



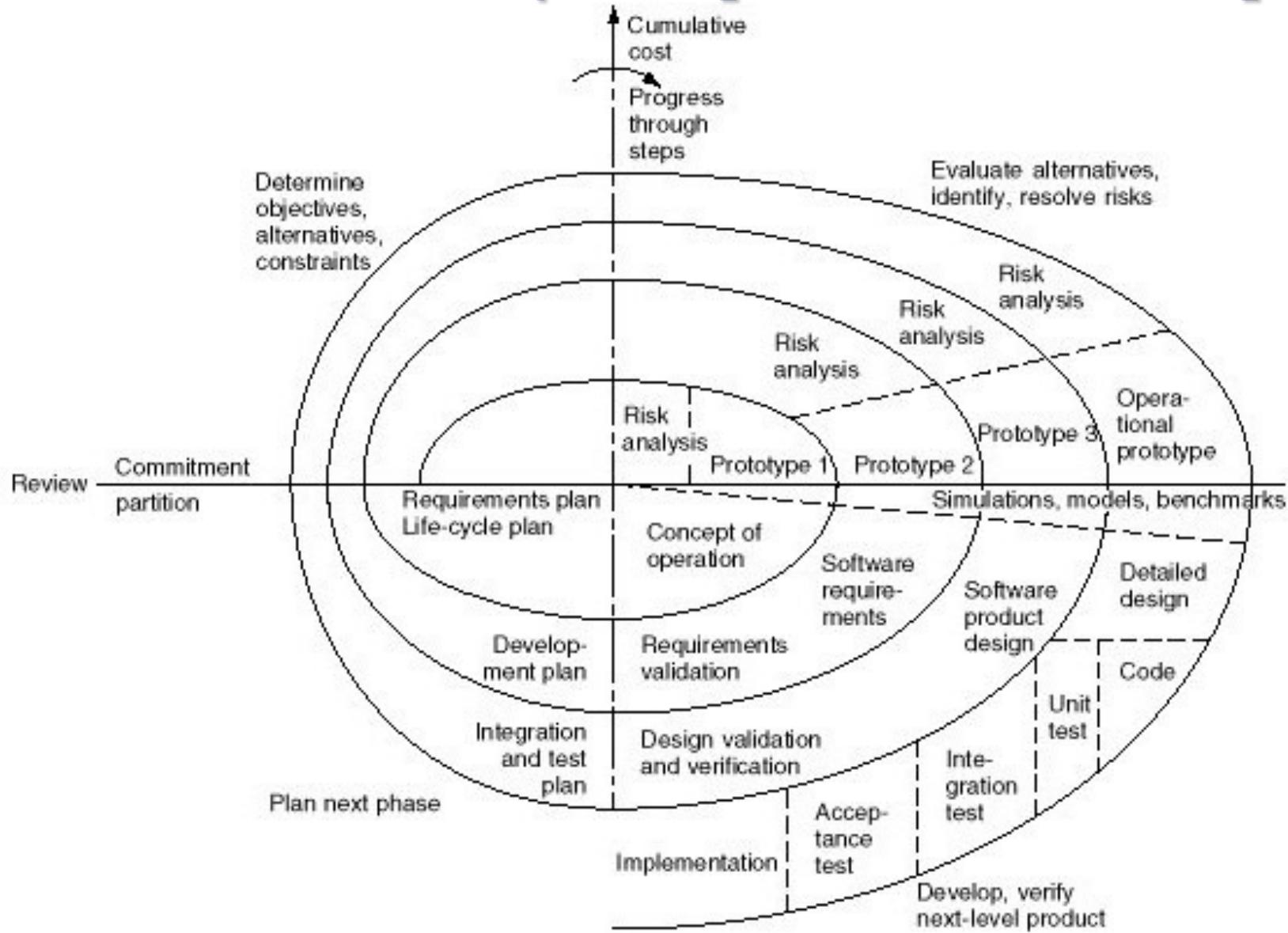
Modello a spirale semplificato (versione lineare)



Modello a spirale semplificato



Modello full-spiral [Boehm, 1988]



Risk management

- Risk management is concerned with identifying risks and drawing up plans to minimise their effect on a project
- A *risk* is a *probability* that some adverse circumstance will occur
- Categories of risk
 - *Project* risks affect schedule or resources
 - *Product* risks affect the quality or performance of the software being developed
 - *Business* risks affect the organisation developing or procuring the software

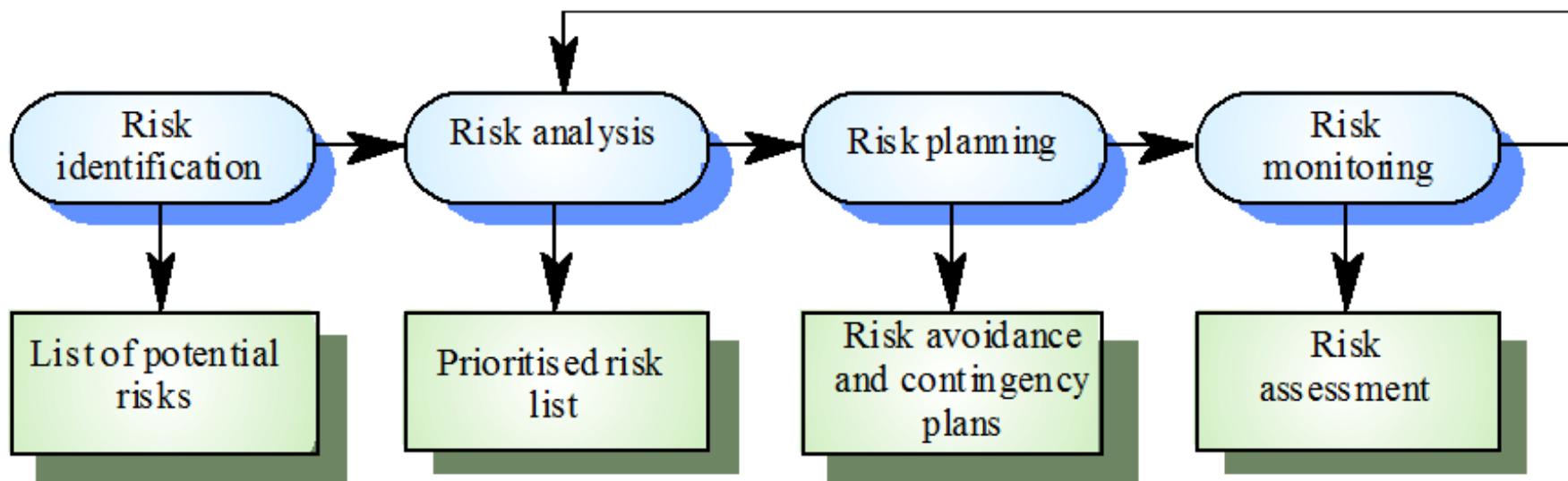
Risks by category

Risk	Risk type	Description
Staff turnover	Project	Experienced staff will leave the project before it is finished.
Management change	Project	There will be a change of organisational management with different priorities.
Hardware unavailability	Project	Hardware which is essential for the project will not be delivered on schedule.
Requirements change	Project and product	There will be a larger number of changes to the requirements than anticipated.
Specification delays	Project and product	Specifications of essential interfaces are not available on schedule
Size underestimate	Project and product	The size of the system has been underestimated.
CASE tool under-performance	Product	CASE tools which support the project do not perform as anticipated
Technology change	Business	The underlying technology on which the system is built is superseded by new technology.
Product competition	Business	A competitive product is marketed before the system is completed.

The risk management process

- Risk identification
 - Identify project, product and business risks
- Risk analysis
 - Assess the likelihood and consequences of these risks
- Risk planning
 - Draw up plans to avoid or minimise the effects of the risk
- Risk monitoring
 - Monitor the risks throughout the project

The risk management process (2)



Risk identification (1)

Risk types

- Technology risks
- People risks
- Organisational risks
- Tools risks
- Requirements risks
- Estimation risks

Risk identification (2)

Risk type	Possible risks
Technology	<p>The database used in the system cannot process as many transactions per second as expected.</p> <p>Software components which should be reused contain defects which limit their functionality.</p>
People	<p>It is impossible to recruit staff with the skills required.</p> <p>Key staff are ill and unavailable at critical times.</p> <p>Required training for staff is not available.</p>
Organisational	<p>The organisation is restructured so that different management are responsible for the project.</p> <p>Organisational financial problems force reductions in the project budget.</p>
Tools	<p>The code generated by CASE tools is inefficient.</p> <p>CASE tools cannot be integrated.</p>
Requirements	<p>Changes to requirements which require major design rework are proposed.</p> <p>Customers fail to understand the impact of requirements changes.</p>
Estimation	<p>The time required to develop the software is underestimated.</p> <p>The rate of defect repair is underestimated.</p> <p>The size of the software is underestimated.</p>

Risk analysis

(1)

- Assess probability and seriousness of each risk
- Risk *probability* may be:
 - very low (<10%)
 - low (10-25%)
 - moderate (25-50%)
 - high (50-75%)
 - very high (>75%)
- Risk *effects* might be catastrophic, serious, tolerable or insignificant

Risk analysis (2)

Risk	Probability	Effects
Organisational financial problems force reductions in the project budget.	Low	Catastrophic
It is impossible to recruit staff with the skills required for the project.	High	Catastrophic
Key staff are ill at critical times in the project.	Moderate	Serious
Software components which should be reused contain defects which limit their functionality.	Moderate	Serious
Changes to requirements which require major design rework are proposed.	Moderate	Serious
The organisation is restructured so that different management are responsible for the project.	High	Serious
The database used in the system cannot process as many transactions per second as expected.	Moderate	Serious
The time required to develop the software is underestimated.	High	Serious
CASE tools cannot be integrated.	High	Tolerable
Customers fail to understand the impact of requirements changes.	Moderate	Tolerable
Required training for staff is not available.	Moderate	Tolerable
The rate of defect repair is underestimated.	Moderate	Tolerable
The size of the software is underestimated.	High	Tolerable
The code generated by CASE tools is inefficient.	Moderate	Insignificant

Risk analysis

(3)

- Identify e.g., the *top-ten risks* by considering:
 - all *catastrophic* risks
 - all *serious* risks that have more than a *moderate* probability of occurrence
- Rank such risks by order of importance

Risk planning

- Consider each risk and develop a strategy to manage that risk
- Avoidance strategies
 - The probability that the risk will arise is reduced
- Minimisation strategies
 - The impact of the risk on the project or product will be reduced
- Contingency plans
 - If the risk arises, contingency plans are strategies to deal with that risk

Risk management strategies

Risk	Strategy
Organisational financial problems	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Recruitment problems	Alert customer of potential difficulties and the possibility of delays, investigate buying-in components.
Staff illness	Reorganise team so that there is more overlap of work and people therefore understand each other's jobs.
Defective components	Replace potentially defective components with bought-in components of known reliability.
Requirements changes	Derive traceability information to assess requirements change impact, maximise information hiding in the design.
Organisational restructuring	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Database performance	Investigate the possibility of buying a higher-performance database.
Underestimated development time	Investigate buying in components, investigate use of a program generator.

Risk monitoring (1)

- Assess each identified risks regularly to decide whether or not it is becoming less or more probable
- To perform assessment look at ***risk factors*** (see next slide)
- Also assess whether the effects of the risk have changed (in such case go back to risk analysis)
- Each key risk should be discussed at management progress meetings

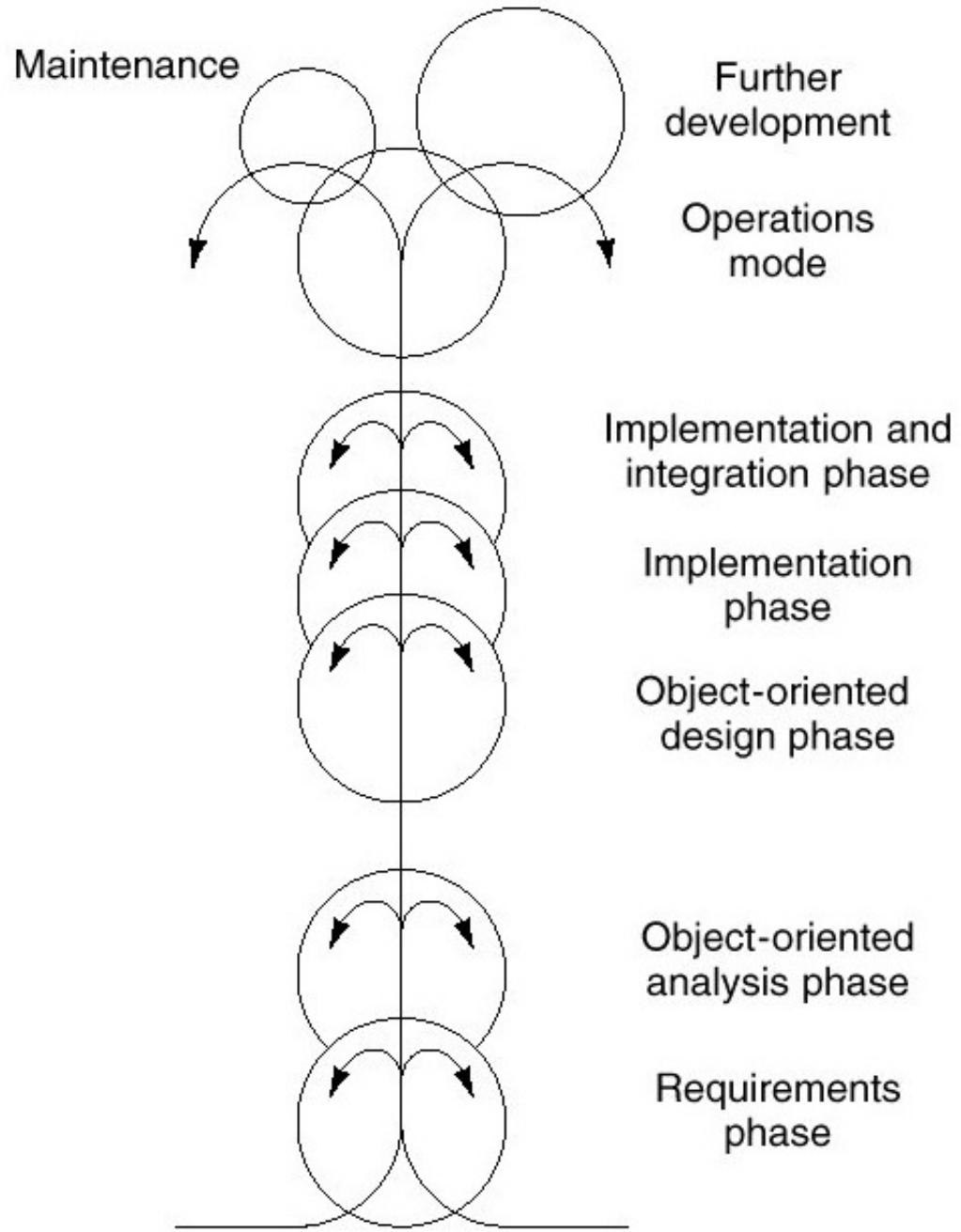
Risk monitoring (2)

Risk factors

Risk type	Potential indicators
Technology	Late delivery of hardware or support software, many reported technology problems
People	Poor staff morale, poor relationships amongst team member, job availability
Organisational	organisational gossip, lack of action by senior management
Tools	reluctance by team members to use tools, complaints about CASE tools, demands for higher-powered workstations
Requirements	many requirements change requests, customer complaints
Estimation	failure to meet agreed schedule, failure to clear reported defects

Altri modelli (1)

Modello object-oriented



Altri modelli (2)

- **Modello di *ingegneria simultanea* (o concorrente)**
 - ha come obiettivo la riduzione di tempi e costi di sviluppo, mediante un approccio sistematico al progetto integrato e concorrente di un prodotto software e del processo ad esso associato.
 - Le fasi di sviluppo coesistono invece di essere eseguite in sequenza.
- **Modello basato su *metodi formali***
 - comprende una serie di attività che conducono alla specifica formale matematica del software, al fine di eliminare ambiguità, incompletezze ed inconsistenze e facilitare la verifica dei programmi mediante l'applicazione di tecniche matematiche.
 - La ***Cleanroom Software Engineering*** (1987) ne rappresenta un esempio di realizzazione, in cui viene enfatizzata la possibilità di rilevare i difetti del software in modo più tempestivo rispetto ai modelli tradizionali