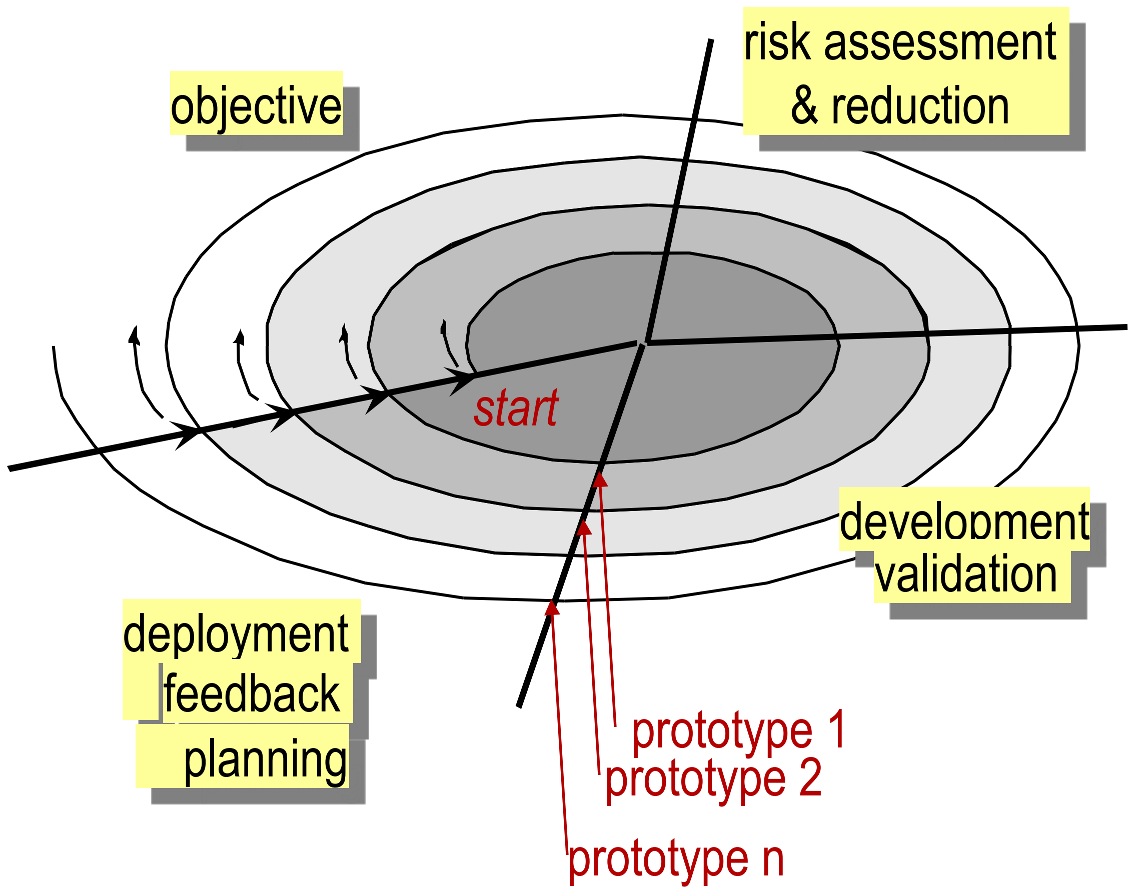
Roadmap

- SE process management
  - Waterfall model
  - Incremental methods
  - Agile/XP methods
  - Iterative / spiral methods (eg, RUP)
  - Evolutionary methods
  - V-Model
- CMMI

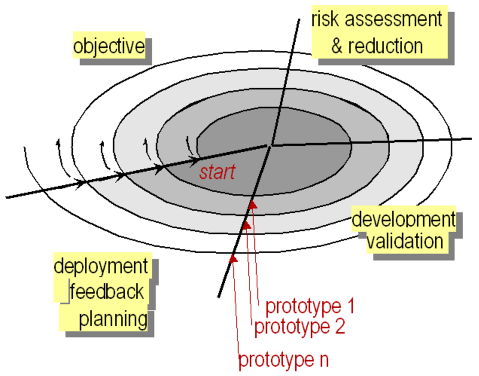
Spiral Model



- Objective setting
  - Identify objectives for this phase
- Risk assessment & reduction
  - Risks assessed
  - activities to reduce key risks
- Development & validation
  - Choose any development model
- Planning
  - Review
  - Plan next spiral phase

Spiral Model: The Rules

- Process is spiral rather than sequence with backtracking
- loop in spiral = one phase in the process
- No fixed phases
  - …such as specification or design
  - loops in spiral chosen depending on what is required
- Risks explicitly assessed & resolved throughout the process

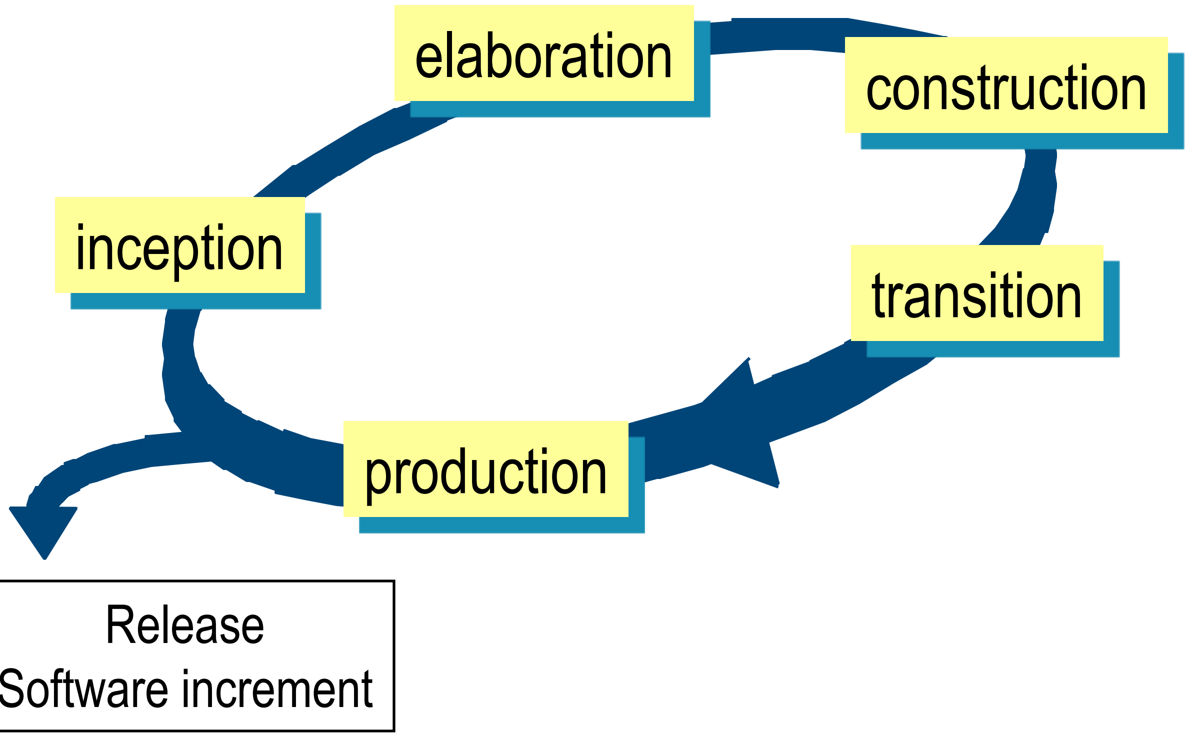


- *Probably suitable for small/medium size high-risk, high-change projects*

[Rational] Unified Process (RUP) Model

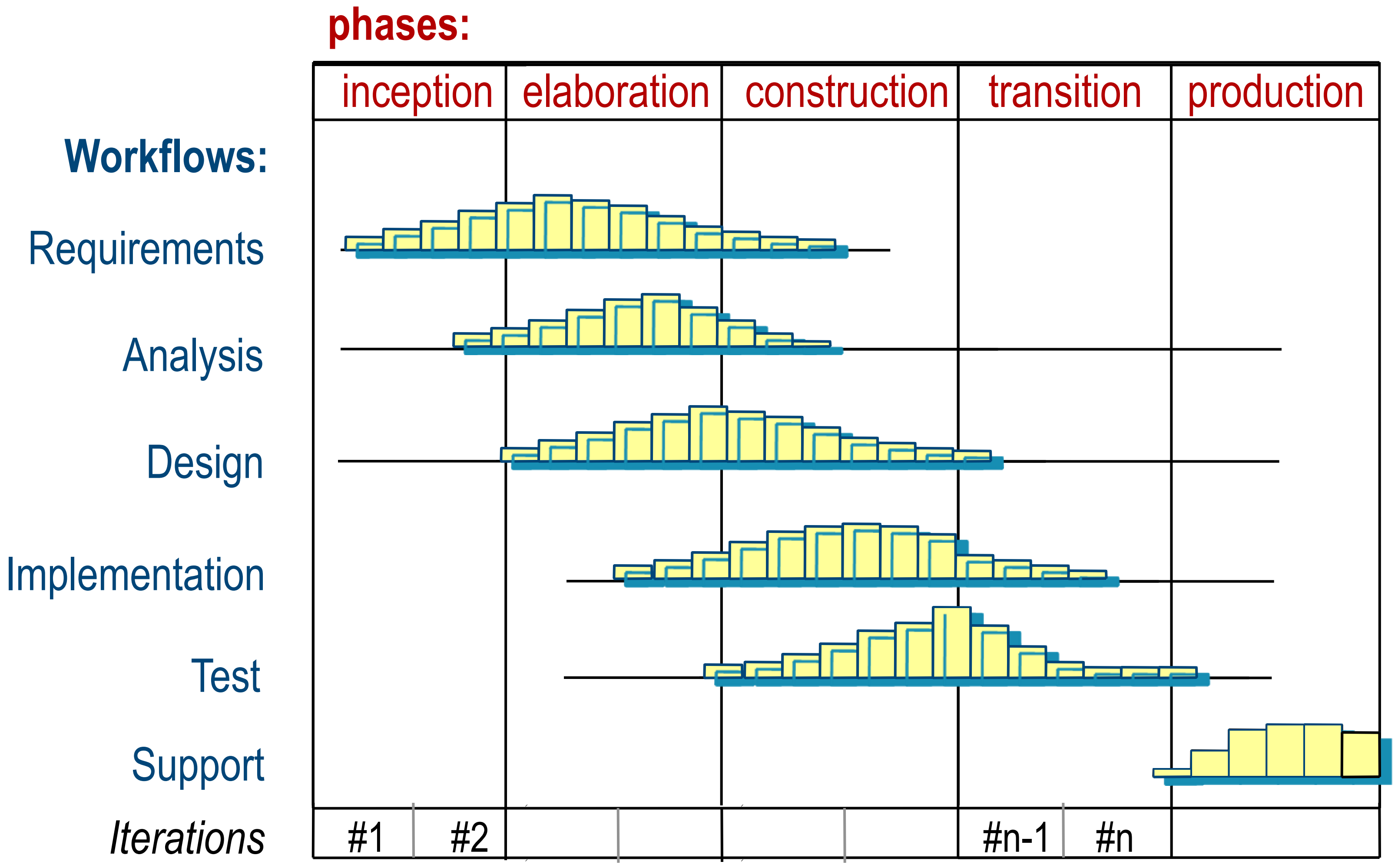
- Software process based on Unified Modeling Language (UML)
  - use-case driven — *…supports req-to-spec transition*
  - architecture-centric — *…supports factorization/modularization*
  - iterative and incremental — *…supports project management*
- Normally described from 3 perspectives:
  - dynamic perspective: phases over time
  - static perspective: process activities
  - practice perspective: suggests good practice

UP Phases



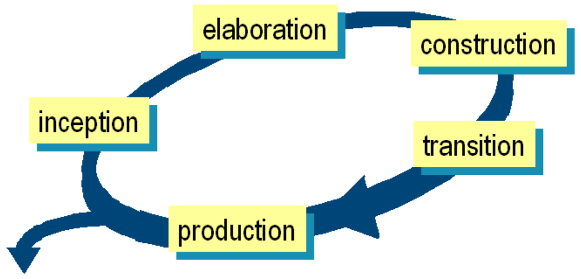
- Inception
  - Establish business case
- Elaboration
  - understanding of problem domain & system architecture
- Construction
  - System design, programming, testing
- Transition
  - Deploy system in operative environment
- Production
  - Support & maintain

UP Phases


phases:
inception
elaboration
construction
transition
production
Workflows:
Requirements
Analysis
Design
Implementation
Test
Support
Iterations
#1
#2
#n-1
#n

UP Key Points

- Rational Unified Process: increased flexibility by combining
  - generic process model
  - incremental ideas
  - evolutionary ideas (prototyping! see next)
- Core characteristic:
  separates activities from phases



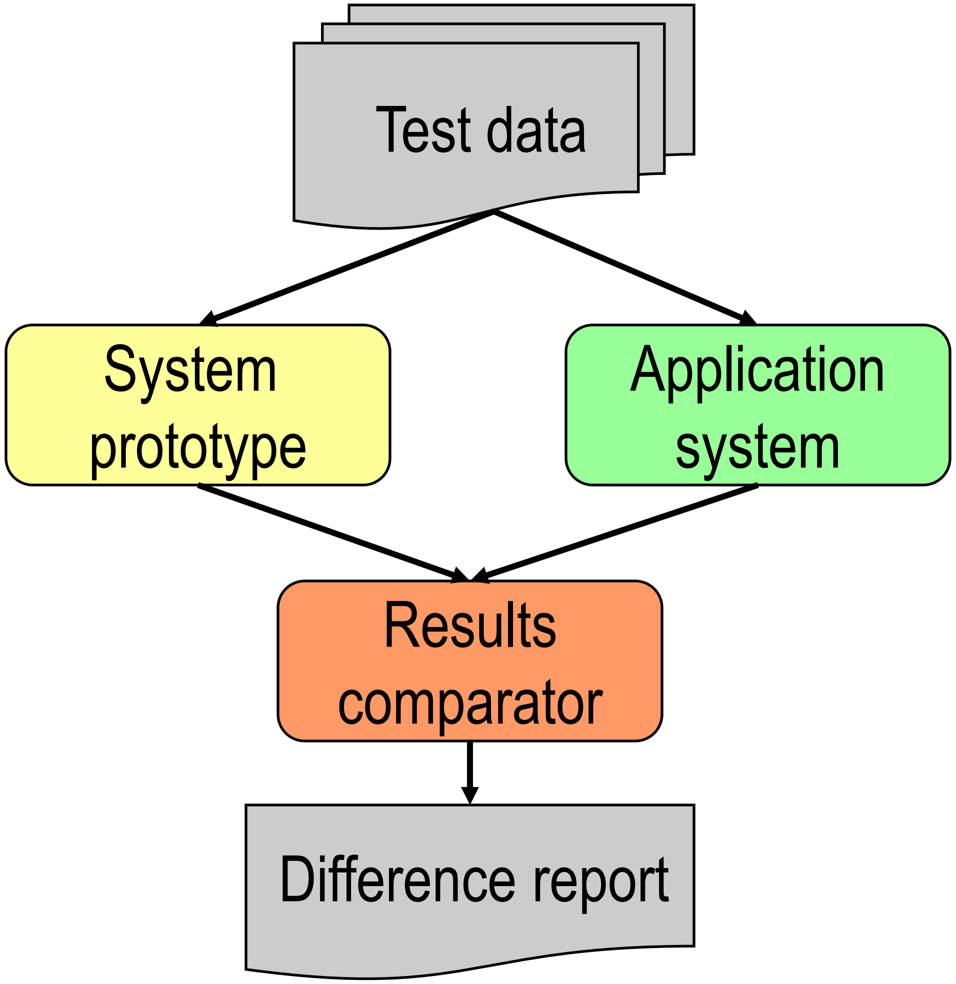
Evolutionary Development

- Exploratory development
  - work with customers
  - evolve final system from initial outline specification
  - *start with well-understood requirements, add new features as proposed by customer*
    $\rightarrow$ similar to incremental / iterative approach
- Throw-away prototyping
  - Goal: understand system requirements,
    not to build a deliverable
  - *start with poorly understood requirements to clarify what is really needed*

Prototyping

- For some large systems, incremental development & delivery may be impractical
  - especially true when multiple teams working on different sites
- Alternative: Prototyping
  - experimental system developed as basis for formulating requirements
  - thrown away when system specification agreed
- prototype = initial version of a system used to
  - demonstrate concepts
  - try out design options
- prototype can be used in:
  - requirements engineering process $\rightarrow$ help with requirements elicitation & validation
  - design processes $\rightarrow$ explore options, develop UI design
  - testing process $\rightarrow$ run back-to-back tests

Back-to-Back Testing



Throw-Away Prototypes

- Prototypes should be discarded after development as they are not a good basis for a production system:
  - may be impossible to tune the system to meet non-functional requirements
  - Prototypes normally undocumented
  - prototype structure usually degraded through rapid change
  - prototype probably will not meet normal organisational quality standards

When Incremental Dev, When Prototype?

- incremental development*:* deliver working system to end-users
  - development starts with requirements best understood
- throw-away prototyping: validate or derive system requirements
  - prototyping process starts with requirements poorly understood

Evolutionary Development: Appraisal

- Problems
  - Lack of process visibility
  - Systems are often poorly structured
  - Special skills (e.g. in languages for rapid prototyping) may be required
- Applicability
  - For small or medium-size interactive systems
  - For well isolated parts of large systems (e.g. the user interface)
  - For short-lifetime systems

Roadmap

- SE process management
  - Waterfall model
  - Incremental methods
  - Agile/XP methods
  - Iterative / spiral methods (eg, RUP)
  - Evolutionary methods
  - V-Model
- CMMI

Vorgehens-Modell

- = Development Standard for IT Systems of the Federal Republic of Germany
  - German national standard, mandatory for gov (and sometimes industry) procured projects
  - Status: V-Model XT (Feb 2005), see www.v-modell-xt.de
- features
  - process model for planning and realizing development projects
  - considering entire system (!) life cycle
  - responsibilities of each participant ("who" has to do "what" and "when")
  - *...rather detailed*
- Classification: waterfall++, "*can be mapped to UP*"