Лекция 2-2

Модели на софтуерни процеси

Съдържание

- Типове модели
- Предписателни модели на процеси
  - Каскаден (водопад) /*The Waterfall model*/
  - Модел на бързата разработка /*The Rapid Application Development (RAD) model*/
  - Фазови (еволюционни) модели /*Evolutionary process models*/
    - Постъпков (инкрементален) модел
    - Итеративен модел
  - Прототипен модел /*The Prototyping model*/
  - Спираловиден модел /*The spiral model*/

Типове модели

- Описателни
  - Как се разработва софтуера
- Предписателни
  - Как би трябвало да се разработва софтуера

Предписателни модели

- Предписателните модели на процеси дефинират специално множество от дейности, задачи, milestones, и работни продукти, които са необходими за създаването на софтуер с високо качество.
- Предписват
  - Множество от елементи на процес – основни дейности, действия, задачи, работни продукти, механизми за осигуряване на качеството, механизми за управление на промените за всеки проект
  - Работен поток – начина по който елементите на процеса се съотнасят помежду си

Разграничаване на моделите на процеси

- По обратната връзка (feedback)
- Използваните методи за управление/контрол по време на разработването
- Времетраенето на дейностите (Timing of activities)

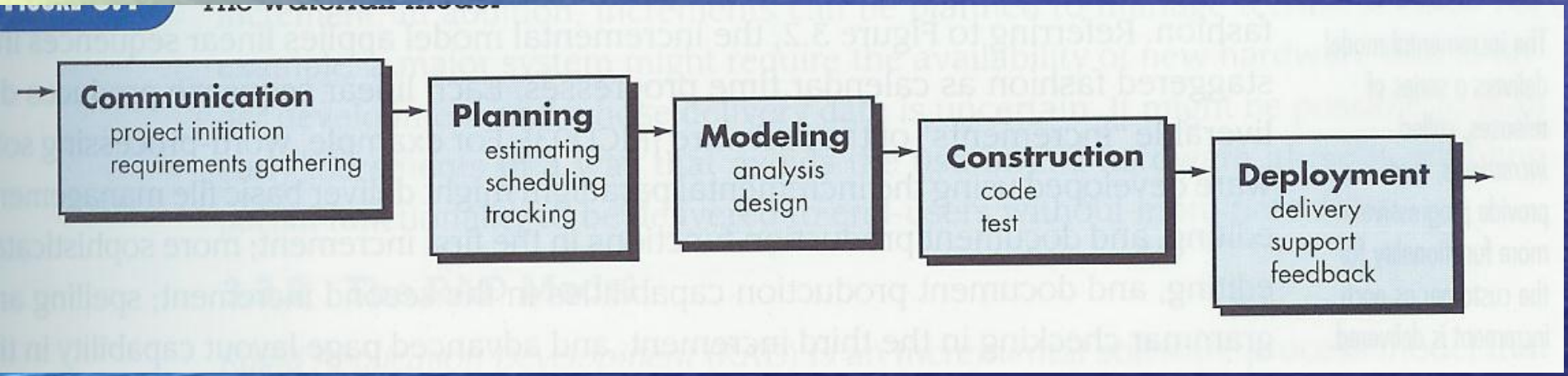
Ad-hoc development

- Възможностите на процеса са непредвидими
- Обикновено графиците, бюджетите, функционалността и качеството на продукта не са съгласувани
- Производителността зависи от способностите на отделните хора и се променя с промяната на техните умения, знания и мотивация

Модел на водопада

- Най-старият метод на структурирано разработване на софтуер
- Предлага систематизиран, последователен подход към разработването на софтуер, който включва следните основни дейности:
  - Събиране на софтуерните изисквания;
  - Оценяване, изготвяне на график, проследяване;
  - Анализ и Проектиране;
  - Генериране на код и Тестване;
  - Доставяне и Поддръжка;

Модел на водопада



Характеристики на модела на водопада

- ясно разграничен процес, който е лесен за разбиране;
- всяка стъпка в модела завършва със създаване на множество от документи
- всяка дейност трябва да бъде напълно завършена, преди да се премине към следваща, а това става, като се одобри множеството от документи;
- ясно са дефинирани входовете и изходите на дейностите, както и интерфейсите между отделните стъпки;
- ясно са дефинирани ролите на разработчиците на софтуер.

Документи на модела на водопада (Sommerville)

Activity	Output documents
Requirements analysis	Feasibility study, Outline requirements
Requirements definition	Requirements document
System specification	Functional specification, Acceptance test plan Draft user manual
Architectural design	Architectural specification, System test plan
Interface design	Interface specification, Integration test plan
Detailed design	Design specification, Unit test plan
Coding	Program code
Unit testing	Unit test report
Module testing	Module test report
Integration testing	Integration test report, Final user manual
System testing	System test report
Acceptance testing	Final system plus documentation

Проблеми на модела на водопада 1

- Реалните проекти рядко следват последователния поток на разработване, който се предлага от модела. Критики:
  - моделът налага по-скоро структура на управление на проект за разработване на софтуер, отколкото да дава насоки как да се извършват отделните дейности;
  - моделът е произлязъл от областта на хардуера и не отчита същността на софтуера като творчески процес на решаване на проблем (с итерации и връщане назад)

Проблеми на модела на водопада 2

- Трудно е за потребителя да формулира всичките си изисквания в началото
- Клиентът трябва да е търпелив
- Разделянето на проекта на отделни етапи не е гъвкаво
- Трудно е да се реагира на променящите се изисквания на клиента

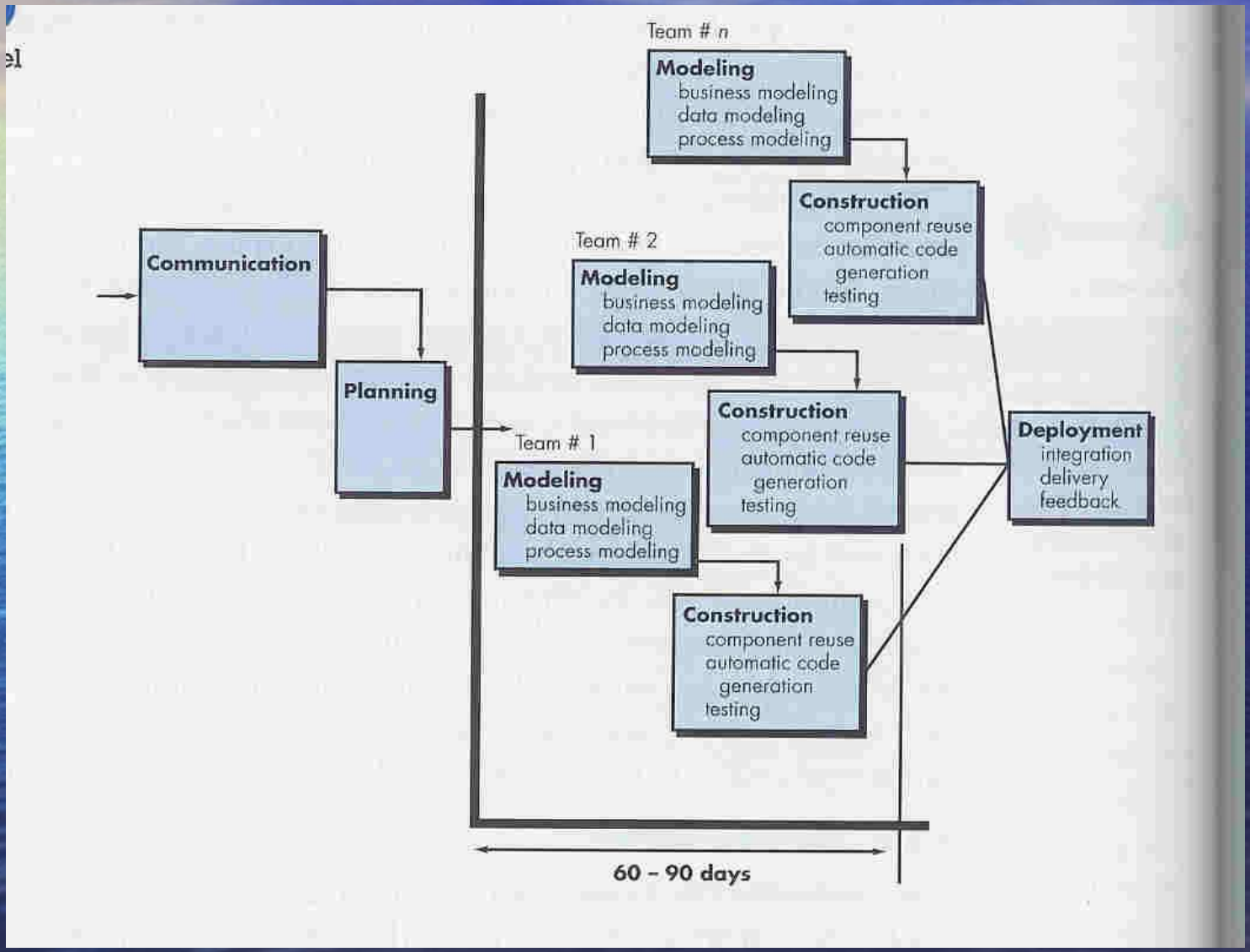
Прилагане на модела на водопада

- Когато изискванията са осъзнати и ясно формулирани в началото
- Когато проектите са ясно организирани - ясно дефинирани роли
- При повторяеми проекти и/или големи проекти, за които времето и бюджетът не са критични

Модел на бързата разработка

- Основна цел - кратък цикъл на разработка
- “Висококоростна” адаптация на модела на водопада
- За постигането на целта се разчита на използването на различни средства за бърза разработка

The RAD model



Дейности на RAD модела

- Комуникация
- Планиране
  - Няколко софтуерни екипа работят паралелно
- Моделиране – паралелно моделиране
  - бизнес моделиране (Business modeling)
  - моделиране на данните (Data modeling)
  - моделиране на процес (Process modeling)
- Конструиране
  - Използване на предварително съществуващи софтуерни компоненти, които се интегрират,
  - Използване на средства за автоматично генериране на код
- Внедряване

Недостатъци на RAD модела

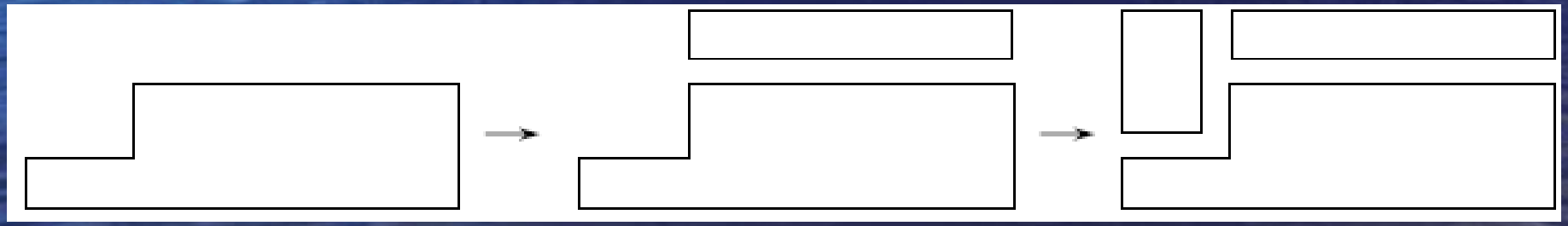
- За големи приложения, подлежащи на разделяне на модули - значителни човешки ресурси
- Когато функционалността на софтуерната система не може да бъде подходящо разделена в отделни модули
- Когато е важна високата производителност на софтуерното приложение
- Когато за разработката на приложението се разчита на все още нови и недостатъчно усвоени технологии

Фазови (еволюционни) модели

- Постъпков (инкрементален) модел
- Итеративен модел

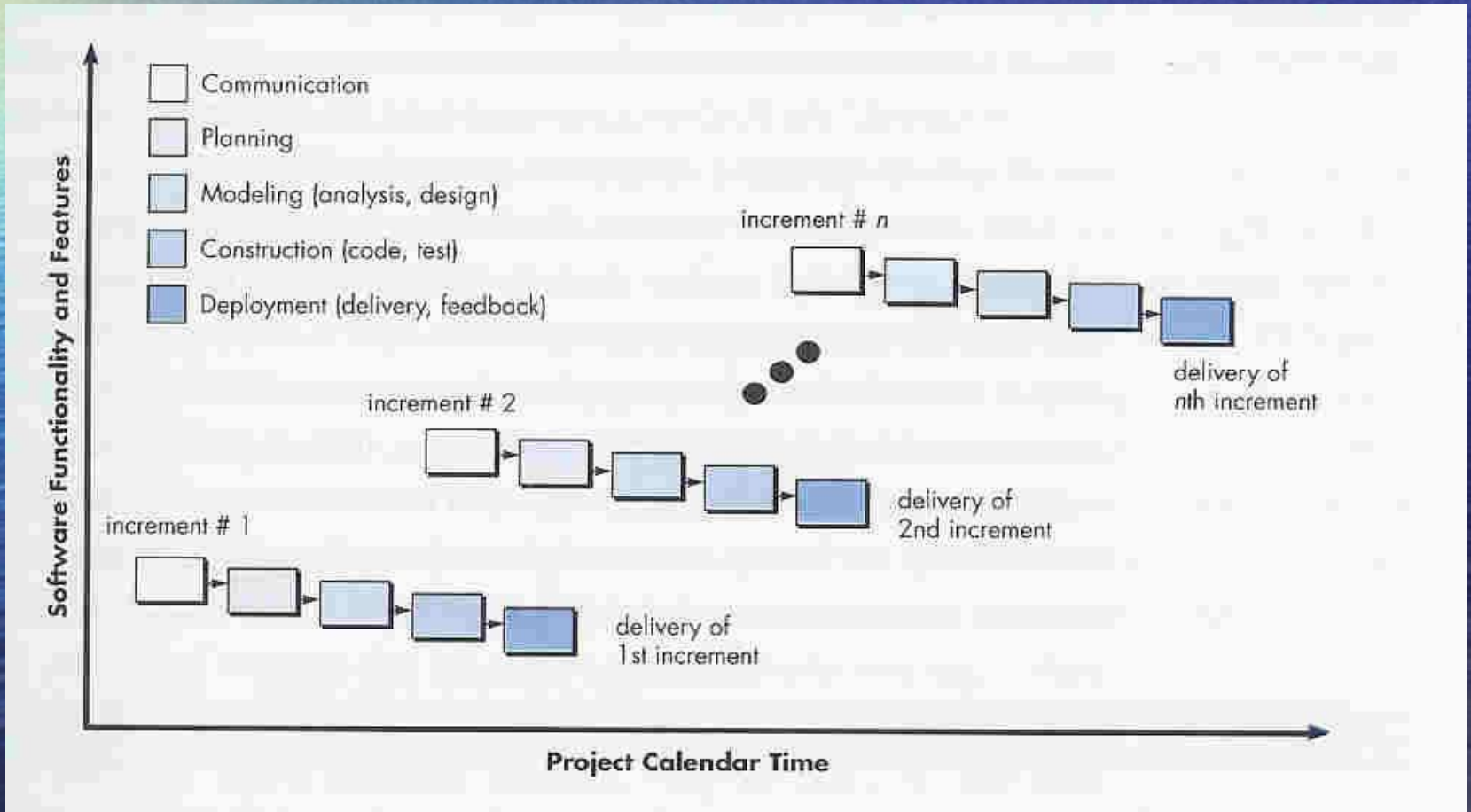
Постъпково разработване

- Системата не се доставя като едно цяло, а вместо това процесът на разработка и доставянето са разделени на стъпки, като всяка стъпка доставя само част от цялата функционалност.
- На идентифицираните потребителски изисквания се присвояват приоритети и тези с по-висок приоритет се реализират в първите стъпки.
- След като започне разработката на една стъпка, изискванията не се променят.



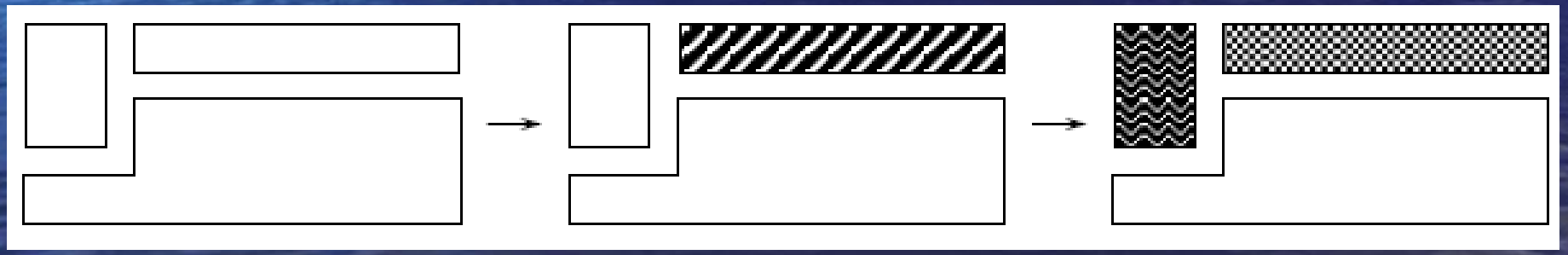
Постъпков (инкрементален) модел

- Комбинира елементи на модела на водопада, но приложени на отделни стъпки



Итеративен модел на процес

- В самото начало доставя цялостната софтуерна система, макар и част от функционалността да е в примитивна форма
- При всяка следваща итерация не се добавя нова функционалност, а само се усъвършенства съществуващата



Проблеми на фазовите (еволюционни) модели

- Необходимостта от активно участие на клиентите по време на изпълнение на проекта може да доведе до закъснения.
- Уменията за комуникация и координация са от особено голямо значение при разработката и ако не са на достатъчно добро ниво, водят до проблеми.
- Неформалните заявки за подобрения след завършването на всяка стъпка могат да доведат до объркване.
- Този модел може да доведе до т. нар. “scope creep” – бавно и постепенно разширяване на обхвата на приложението, без процесът да е сходящ

Предимства на фазовите (еволюционни) модели

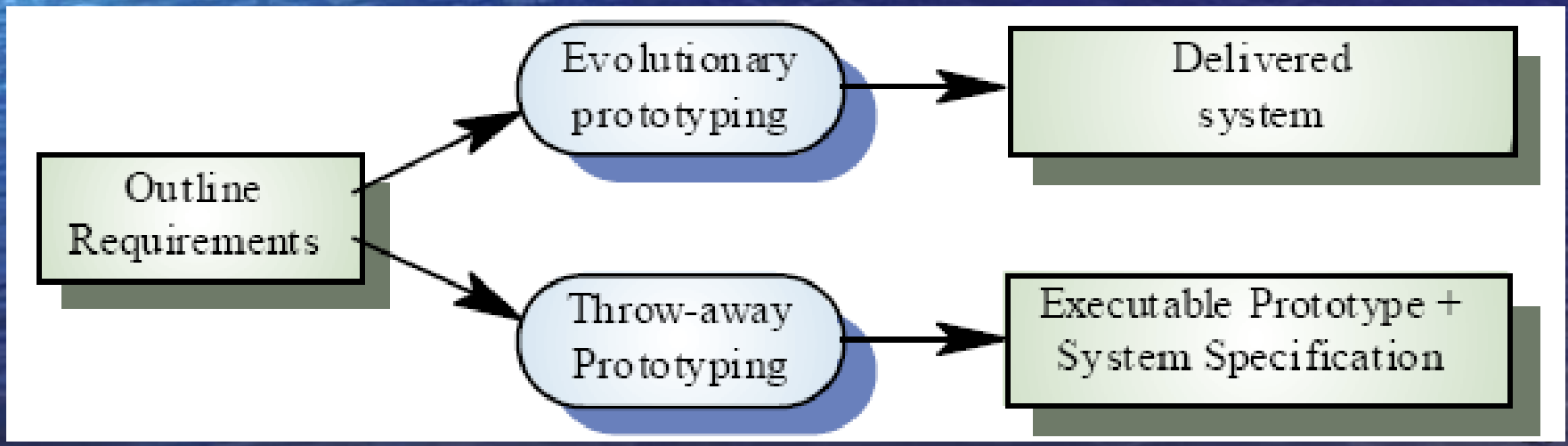
- Клиентът може да използва системата, преди да е готов целият продукт.
- Първите стъпки могат да служат като прототип, за да помогнат за извличане и изясняване на изискванията към следващите стъпки.
- По-малък риск от неуспех на целия проект.
- Функционалностите от цялата система, които са с най-висок приоритет, са тествани най-много

Прилагане

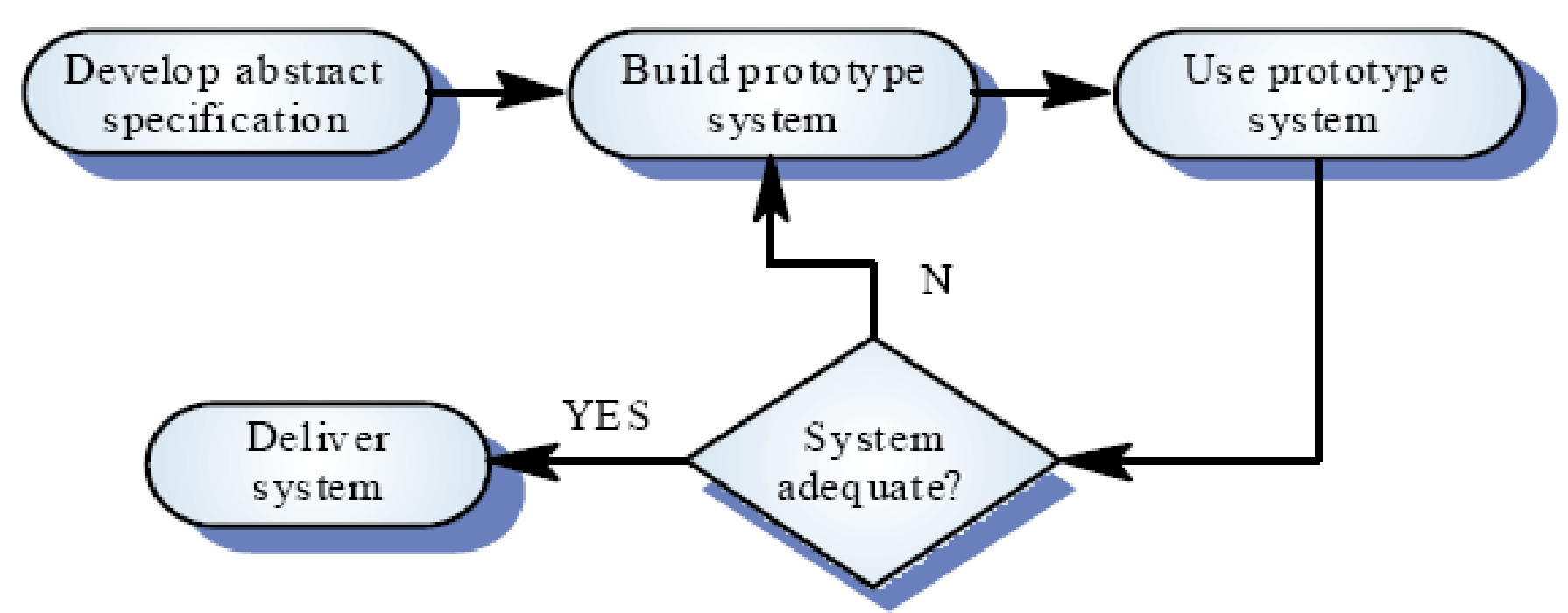
- Когато организацията няма достатъчно човешки ресурс за цялостната реализация в определен срок
  - В разработването на по-ранните версии участват по-малко хора и в зависимост от обратната връзка, получена от клиентите, могат да се присъединят още разработчици на следващите итерации
- Когато с итерациите може да се управляват технологичните рискове
  - Итерация, която изисква използването на нова технология или продукт може да се планира по-късно с цел да има достатъчно време да се усвои или да се достави новият продукт

Прототипен модел

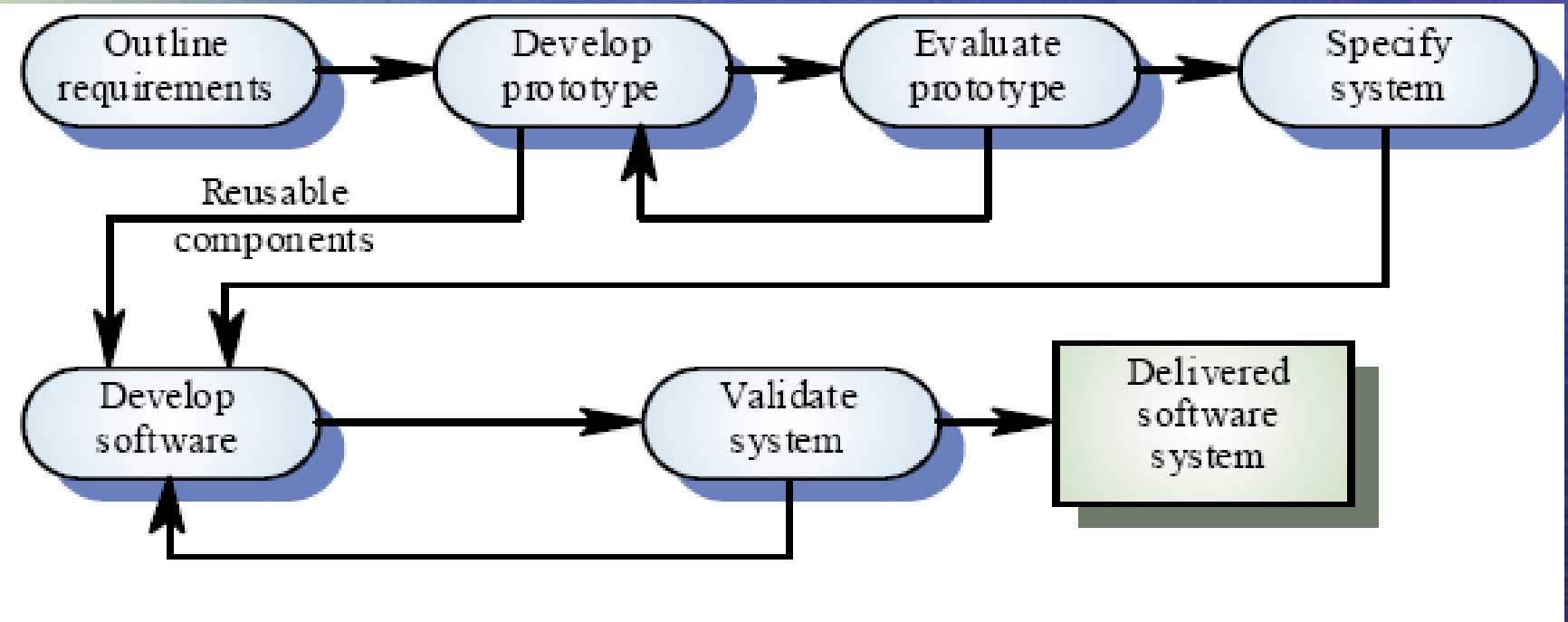
- Два типа
  - Еволюционен прототип
    - Цел - да достави работеща система на крайния потребител
  - Изхвърлен (throw-away) прототип
    - Цел - да подпомогне специфицирането на изискванията към софтуера



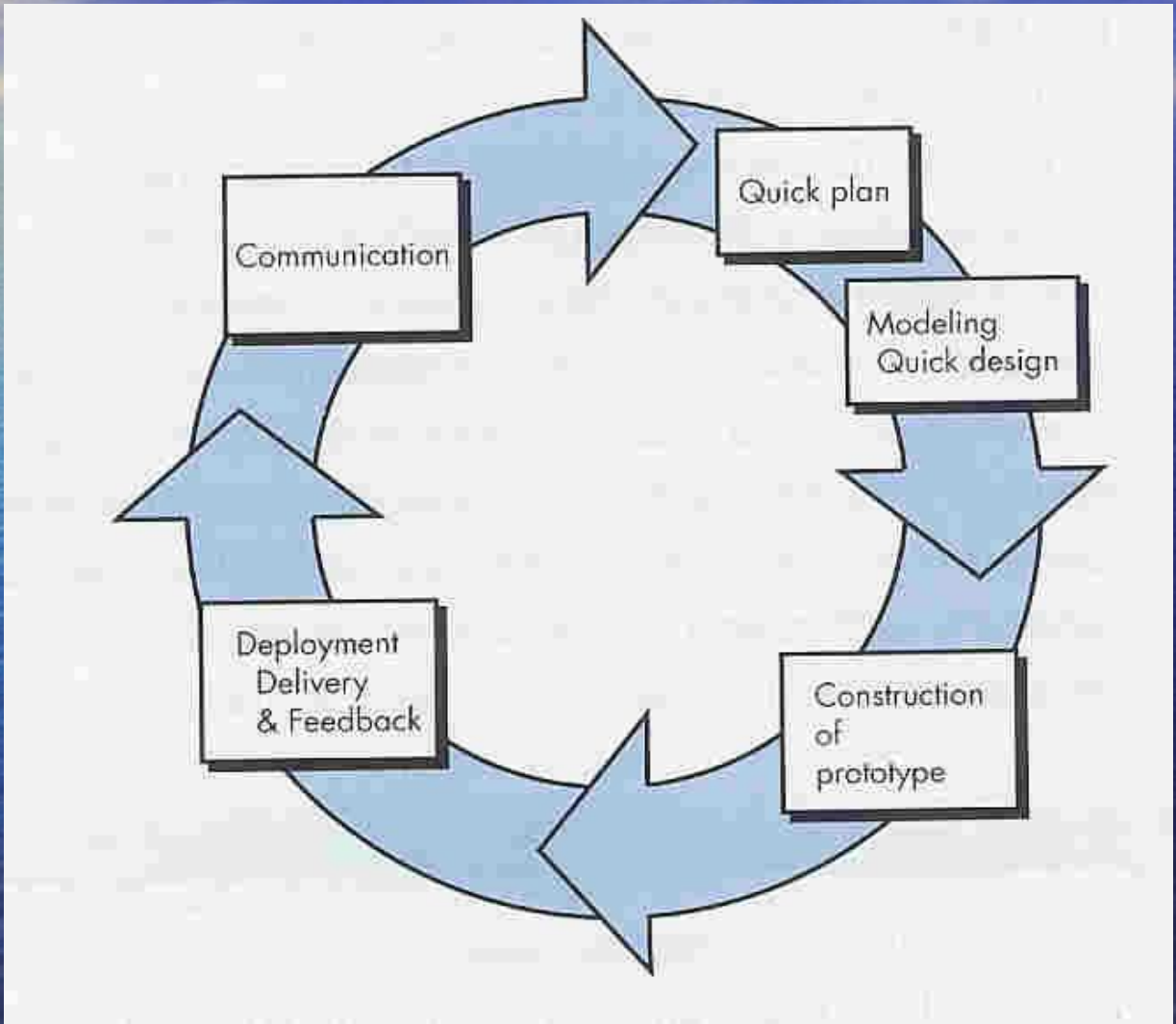
Еволюционен прототип



Изхвърлен (throw-away) прототип



Прототипен модел (Prototyping)



Подходи

- Създаване на основните потребителски интерфейси, без да има някакво значително кодиране.
- Разработване на съкратена версия на системата, която изпълнява ограничено подмножество от функции.
- Използване на съществуваща система или компоненти от система, за да се демонстрират някои функции, които ще бъдат включени в разработваната система

Проблеми на прототипния модел

- Прототипният модел може да използва значителни ресурси, а като резултат прототипът да не успее да удовлетвори очакванията.
- Прототипът може да доведе до лошо проектирана система, ако самият той стане част от крайния продукт.
- Прототипният модел не е подходящ за използване при разработване на софтуерни системи, където
  - проблемът е добре разбран и
  - интерфейсът е ясен и прост

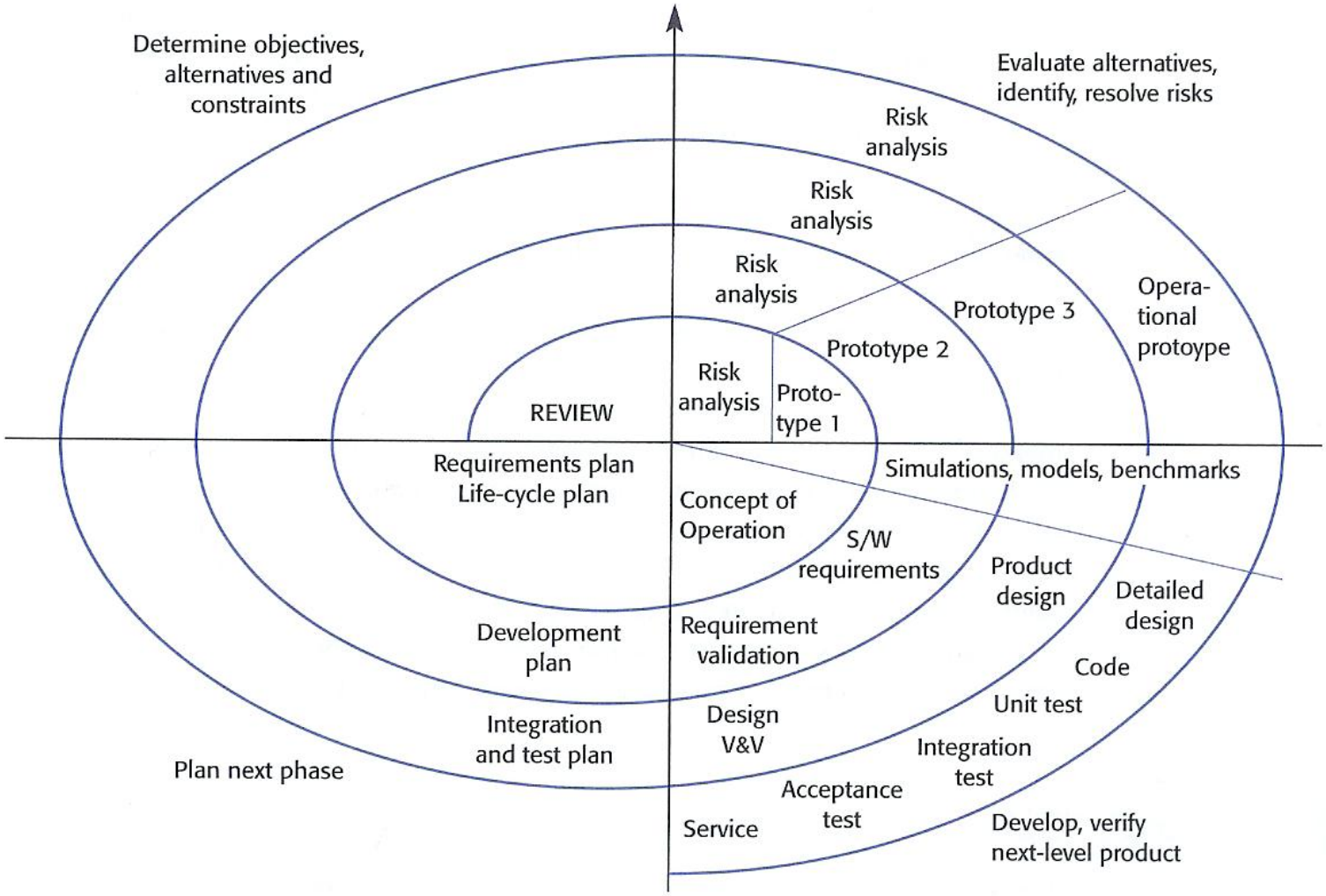
Прилагане

- В проекти, където не са достатъчно ясни потребителските изисквания и дизайнът на софтуерната система
- Както самостоятелно, така и в комбинация с други модели на процеси - модел на водопада, спираловиден модел, постъпков модел и т.н.

Спираловиден модел

- Спираловидният модел е еволюционен модел на софтуерен процес, който съчетава прототипния модел и модела на водопада
- Движещият фактор е **анализ на риска**
- Основни характеристики:
  - итеративен/цикличен подход
  - има множество от точки на прогреса (anchor point milestones)

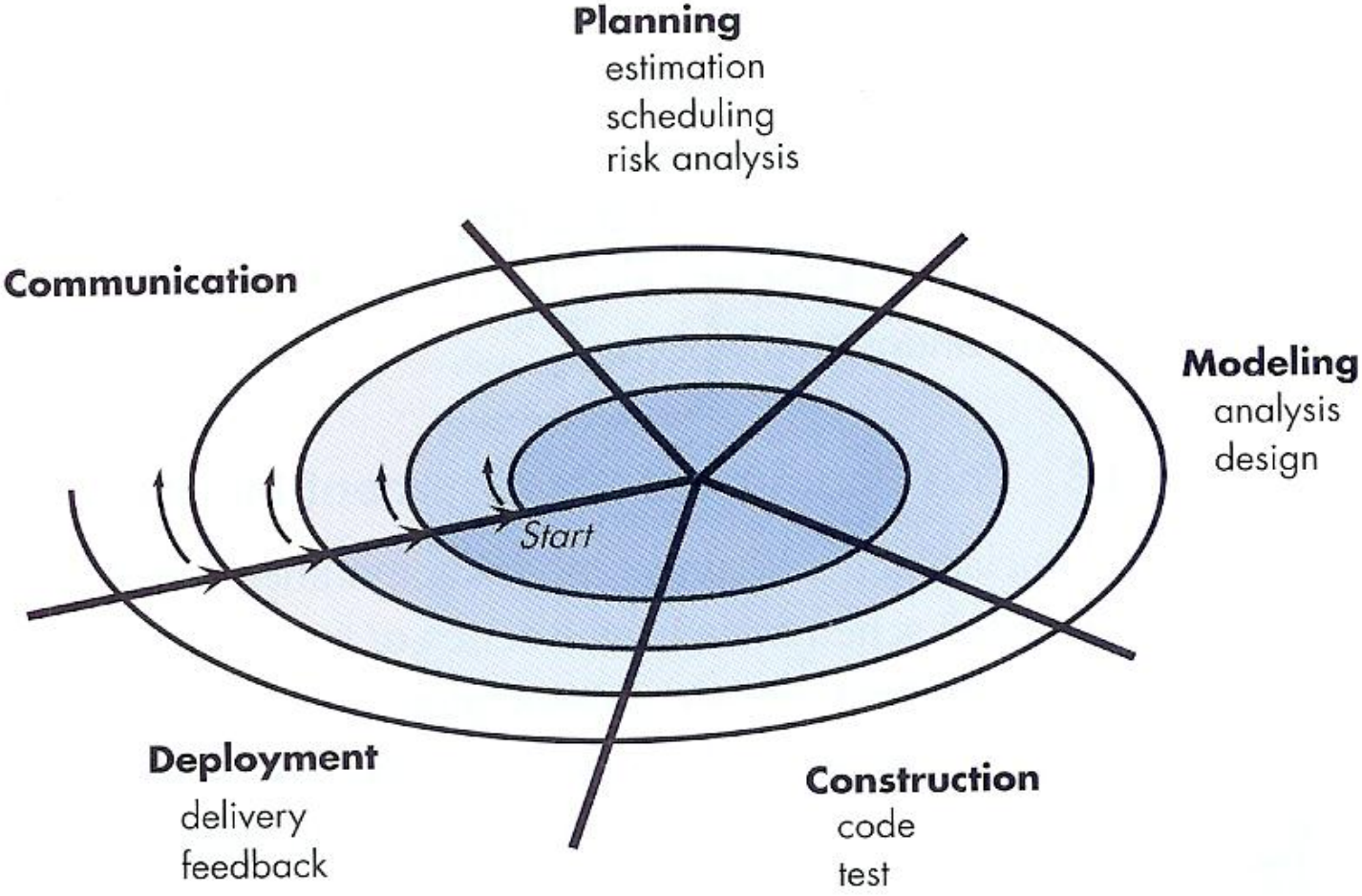
Спираловиден модел



Сектори

- Установяване на целите
  - определят се целите, алтернативите и ограниченията на текущата фаза от разботката;
- Оценка на рисковете и намаляването им
  - идентифицират се и се анализират потенциалните рискове
  - предприемат се действия за намаляването или елиминирането им;
- Разработване и валидиране
  - избира се модел за разработване на текущата фаза;
- Планиране
  - преглежда се и се анализира текущото състояние
  - планира се следващото завъртане по спиралата

Спираловиден модел



Проблеми със спираловидния модел

- Може да се окаже трудно да се убедят клиентите, че процесът на разработка е контролируем, а не е безкраен цикъл.
- Изисква се участието на разработчици с компетентност за оценка на рисковете.
- Ако не се идентифицира и открие някой основен риск, това може да доведе до неуспех.

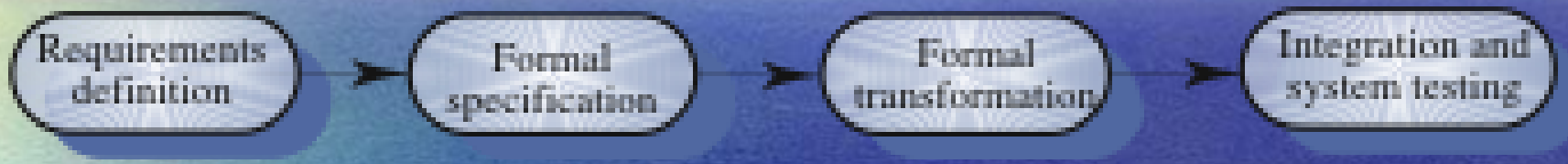
Прилагане

- При разработване на големи (large-scale) софтуерни системи.
- Може да се адаптира и да се прилага през целия жизнен цикъл на софтуера:
  - проекти за изграждане на концепция за софтуера
  - проекти за разработване на нов продукт
  - проекти за подобрение на продукт

Метод на формална трансформация

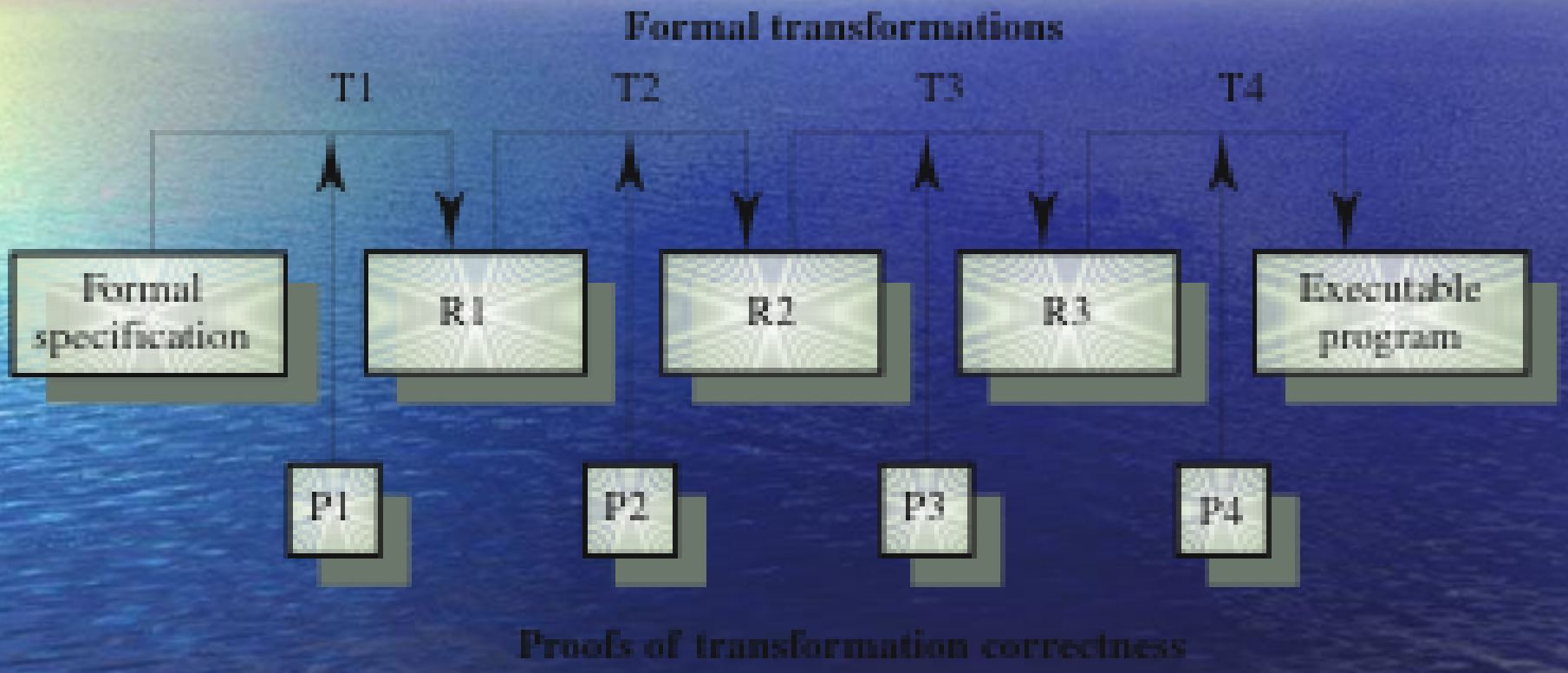
- Основава се на математическо трансформиране на спецификацията на системните изисквания до изпълнима програма
- При трансформирането трябва да се запази коректността и ясно да се покаже, че изпълнимата програма съответства на спецификацията

Метод на формална трансформация - 2



- Спецификацията на софтуерните изисквания се усъвършенства в детайлна формална спецификация, изразена с математическа нотация
- Дейностите моделиране и конструиране са заменени с разработване и прилагане на трансформации

Формални трансформации



Проблеми на модела с формални методи

- разработването на формални модели е скъп и бавен процес;
- необходими са разработчици със специализира-ни умения, както и обучение за това, как да се прилага формалната трансформация;
- поради сложността си изготвените модели трудно могат да се използват;
- някои от аспектите на софтуерна система е трудно е да се специфицират формално – например потребителският интерфейс

Прилагане

- При разработването на софтуерни системи, които са свързани с поддръжката на човешки живот и за които трябва да са гарантирани сигурността и отказоустойчивостта на софтуера, преди да започне реалното му използване

Избор на подходящ модел на процес

- Изборът зависи основно от два фактора:
  - Организационната среда
  - Същността на приложението
- Видове взаимовръзки между информационната система и нейната организационна среда
  - Стабилна среда (unchanging)
  - Променяща се среда (turbulent)
  - Неопределена среда (uncertain)
  - Адаптивна среда (adaptive)

Заключение

- Предписателни модели на софтуерен процес
  - Различен поток на процес
  - Еднакво множество от основни дейности
- Модел на водопада
  - Линейно развитие
  - Приложимост – добре дефинирани и стабилни изисквания
- Модел на бързата разработка
- Инкрементални модели на софтуерен процес
  - Series of increment releases
- Еволюционни модели на софтуерен процес
  - produce incremental work products quickly