



Software Engineering I

Process Models

Dr. Elham Mahmoudzadeh
Isfahan University of Technology
mahmoudzadeh@iut.ac.ir
2022



Software Process models

- Describes the **sequences of SDLC steps**.
- Is a **sequence** of **activities** that leads to the **production** of a software product.

دنباله ای از فعالیت هایی است که منجر به تولید یک محصول نرم افزاری می شود



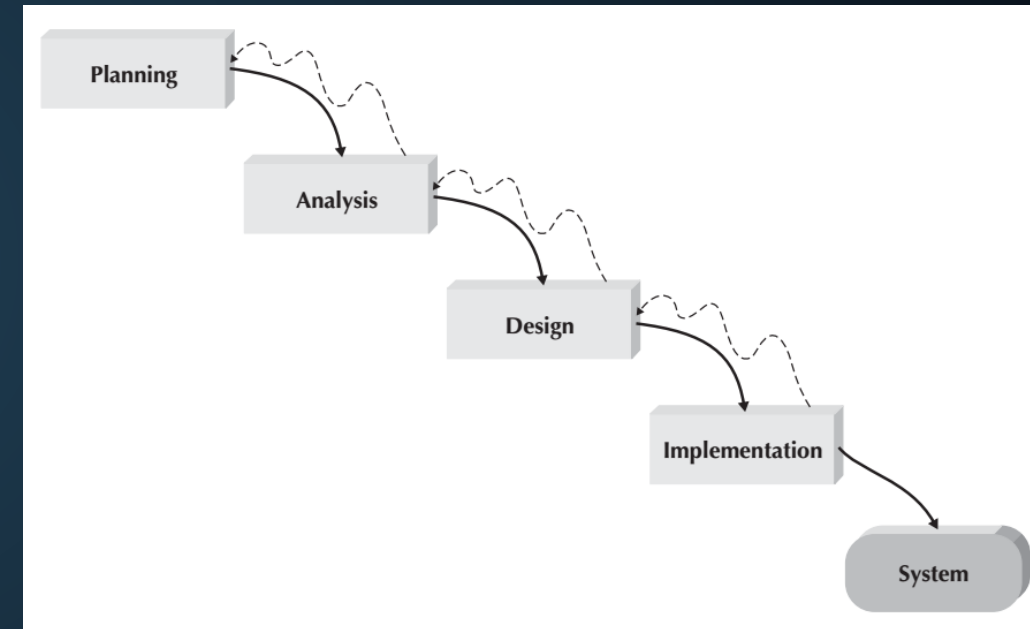
Process Models types

- **Structured**
 - Waterfall
 - parallel
- **RAD**
 - Phased
 - Prototyping
 - Throwaway-prototyping
- **Agile**



Waterfall(I)

- **Proceed in sequence** from one phase to the next.
- The key deliverables for each phase are typically **very long** (often hundreds of pages in length) and are presented to the **project sponsor** for approval as the project moves from phase to phase. **Once** the sponsor **approves** the work that was conducted for **a phase**, the phase **ends** and the next one begins.
- It moves forward **from phase to phase** in the same manner as a waterfall. Although it is possible to go backward in the SDLC (e.g., from design back to analysis), it is **extremely difficult**.
- Because of the **cascade** from one phase to another, this model is known as the 'waterfall model.'





Waterfall(II)

طراحی باید قبل از شروع برنامه نویسی به طور کامل مشخص شود

زمان زیادی بین تکمیل پیشنهاد سیستم در مرحله تجزیه و تحلیل و تحویل سیستم (معمولاً چندین ماه یا سال) می گذرد

نیازهای سیستم را مدتها قبل از شروع برنامه نویسی شناسایی می کند
این تغییرات در الزامات را در طول پروژه به حداقل می رساند

- Key **advantages**

- It identifies system **requirements long before** programming begins.
- It **minimizes changes** to the requirements as the project proceeds.

- Key **disadvantages**

- **Design** must be **completely specified** before programming begins
- **A long time elapses** between the completion of the system **proposal** in the analysis phase and the **delivery** of the system (usually many months or years).
- If the project team **misses** important requirements, **expensive post-implementation** programming may be needed. A system can also require significant rework because the business environment has changed from the time when the analysis phase occurred.

اگر تیم پروژه الزامات مهم را از دست بدهد، ممکن است به برنامه ریزی گرانقیمت پس از پیاده سازی نیاز باشد. یک سیستم همچنین میتواند نیاز به تجدید نظر قابل توجهی داشته باشد، زیرا محیط کسب و کار از زمانی که مرحله تجزیه و تحلیل رخ داده تغییر کرده است



Waterfall Main drawback

اشکال اصلی مدل آبشار دشواری انطباق با تغییرات پس از انجام فرآیند است.
در اصل، یک مرحله باید قبل از رفتن به مرحله بعدی کامل شود.

- The **main drawback** of the waterfall model is the **difficulty of accommodating change after the process is underway.**
- In principle, a phase has to be complete before moving onto the next phase.



Waterfall Model - Usage

در اصل، مدل آبشار تنها زمانی باید مورد استفاده قرار گیرد که الزامات به خوبی درک شده باشند و بعید است که در طول توسعه سیستم به طور اساسی تغییر کنند.

- In principle, the waterfall model should **only be used** when the **requirements are well understood** and **unlikely to change** radically during system development.
- The waterfall model is mostly used for **large systems** engineering projects where a system is developed at **several sites**.
- In those circumstances, the **plan-driven** nature of the waterfall model helps **coordinate** the work.

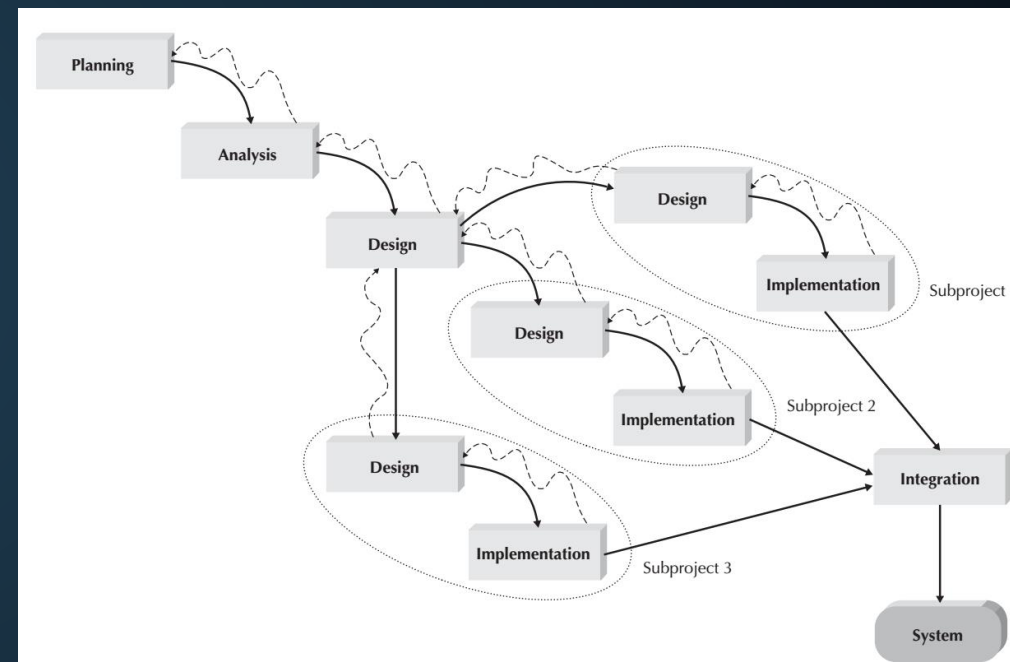
در آن شرایط، ماهیت طرح محور مدل آبشار به هماهنگی کار کمک می کند.



Parallel Development(I)

تلاش برای رسیدگی به مشکل تاخیرهای طولانی بین مرحله تجزیه و تحلیل و تحویل سیستم

- Attempts to address the problem of **long delays** between the analysis phase and the delivery of the system.
- Instead of doing design and implementation in sequence, it performs **a general design** for the **whole system**.
- Then **divides** the project into a series of **distinct subprojects** that can be designed and implemented in **parallel**.
- Once all **subprojects** are **complete**, the separate pieces are **integrated** and the system is delivered.





Parallel Development(III)

- The **primary advantage** is that it can **reduce the time to deliver** a system;
- However, sometimes the subprojects are not **completely independent**; design decisions made in one subproject can affect another.
- At the end of the project, it requires significant **integration efforts**.

مزیت اصلی این است که می تواند زمان تحویل یک سیستم را کاهش دهد.
با این حال، گاهی اوقات پروژه های فرعی کاملاً مستقل نیستند.
تصمیمات طراحی گرفته شده در یک پروژه فرعی می تواند دیگری را تحت تأثیر قرار دهد.
در پایان پروژه، نیاز به تلاش های ادغام قابل توجهی دارد.



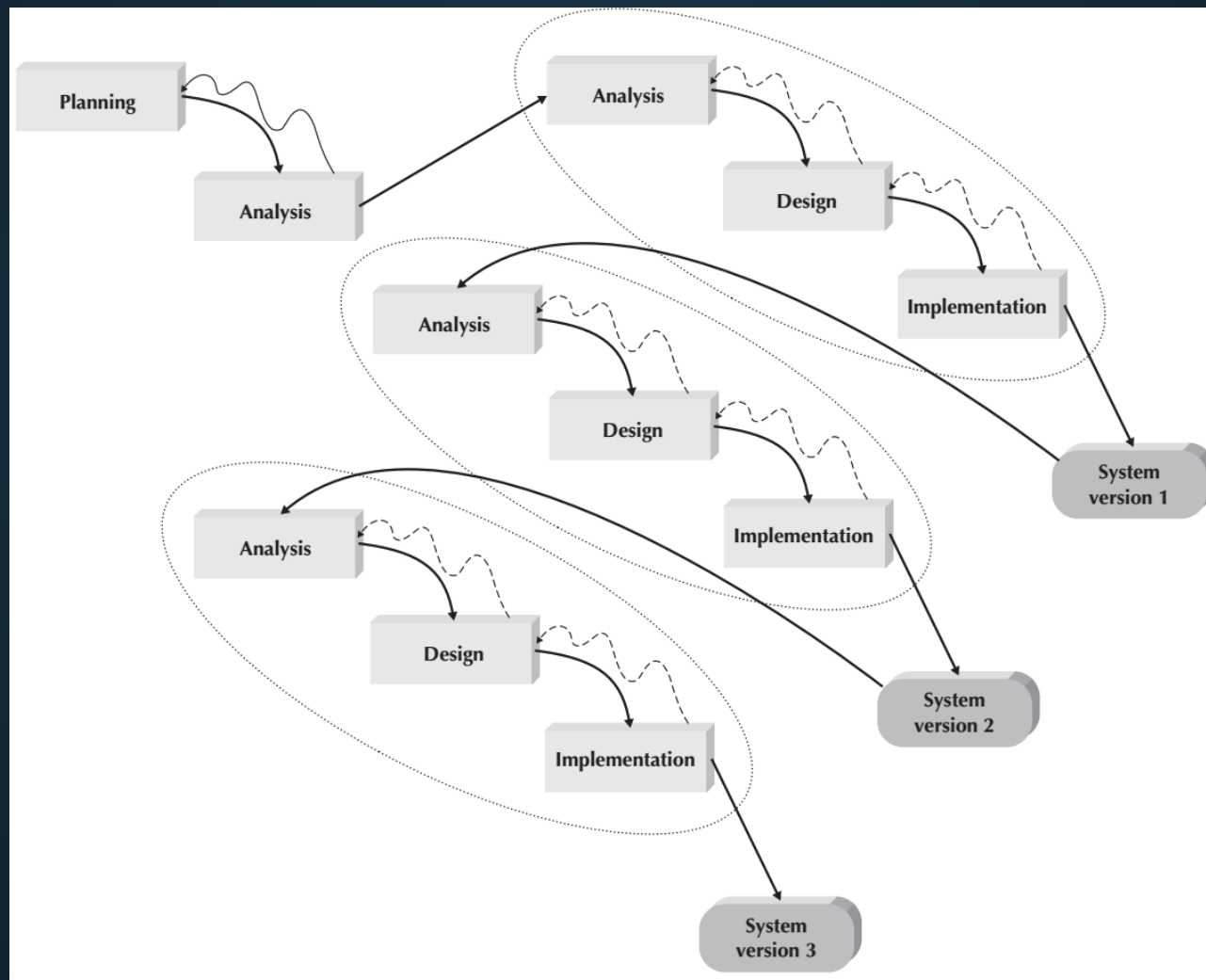
یک سیستم کلی را به مجموعه ای از نسخه هایی که به صورت متوالی توسعه می یابند، تجزیه می کند. مرحله تجزیه و تحلیل مفهوم کلی سیستم را شناسایی می کند و تیم پروژه، کاربران و حامیان سیستم، نیازمندی ها را در یک سری نسخه طبقه بندی می کنند.

Phased Development(I)

- Breaks an **overall system** into a series of *versions* that are developed **sequentially**. The analysis phase identifies the **overall system concept**, and the project team, users, and system sponsor then **categorize the requirements into a series of versions**.
- The **most important** and **fundamental requirements** are bundled into the **first version** of the system.
- The **analysis phase** then leads into design and implementation—but only with **the set of requirements identified** for **version 1**. Once version 1 is implemented, work begins on version 2. **Additional analysis** is performed based on the previously identified requirements and combined with **new ideas and issues** that arose from the **users' experience** with version 1. Version 2 then is designed and implemented, and work immediately begins on the next version. This process continues until the system is complete or is no longer in use.



Phased Development(II)





Phased Development(III)

• اشکال عمده این است که کاربران شروع به کار با سیستم هایی می کنند که عمداً ناقص هستند
شناسایی مهم ترین و مفیدترین ویژگی ها و گنجاندن آنها در نسخه اول و مدیریت انتظارات کاربران در طول مسیر بسیار مهم است

- It has the advantage of **quickly** getting a **useful** system into the **hands** of the **users**.
- Although the system does not perform **all the functions** the users need at first, it does begin to provide **business value** **sooner** than if the system were delivered after completion.
- Likewise, because users begin to work with the system sooner, they are more likely to **identify important additional requirements sooner** than with structured design situations.
- The **major drawback** is that users begin to work with systems that are **intentionally incomplete**. It is critical to identify the most **important** and **useful** features and include them in the **first version** and to manage **users' expectations** along the way.

این مزیت این است که به سرعت یک سیستم مفید را در اختیار کاربران قرار می دهد
اگرچه سیستم در ابتدا تمام عملکردهای مورد نیاز کاربران را انجام نمی دهد، اما زودتر از زمانی که سیستم پس از تکمیل تحویل داده شود، ارزش تجاری را ارائه می کند
به همین ترتیب، از آنجایی که کاربران زودتر شروع به کار با سیستم می کنند، احتمال بیشتری دارد که نیازهای اضافی مهم را زودتر از موقعیت های طراحی ساختاریافته شناسایی کنند

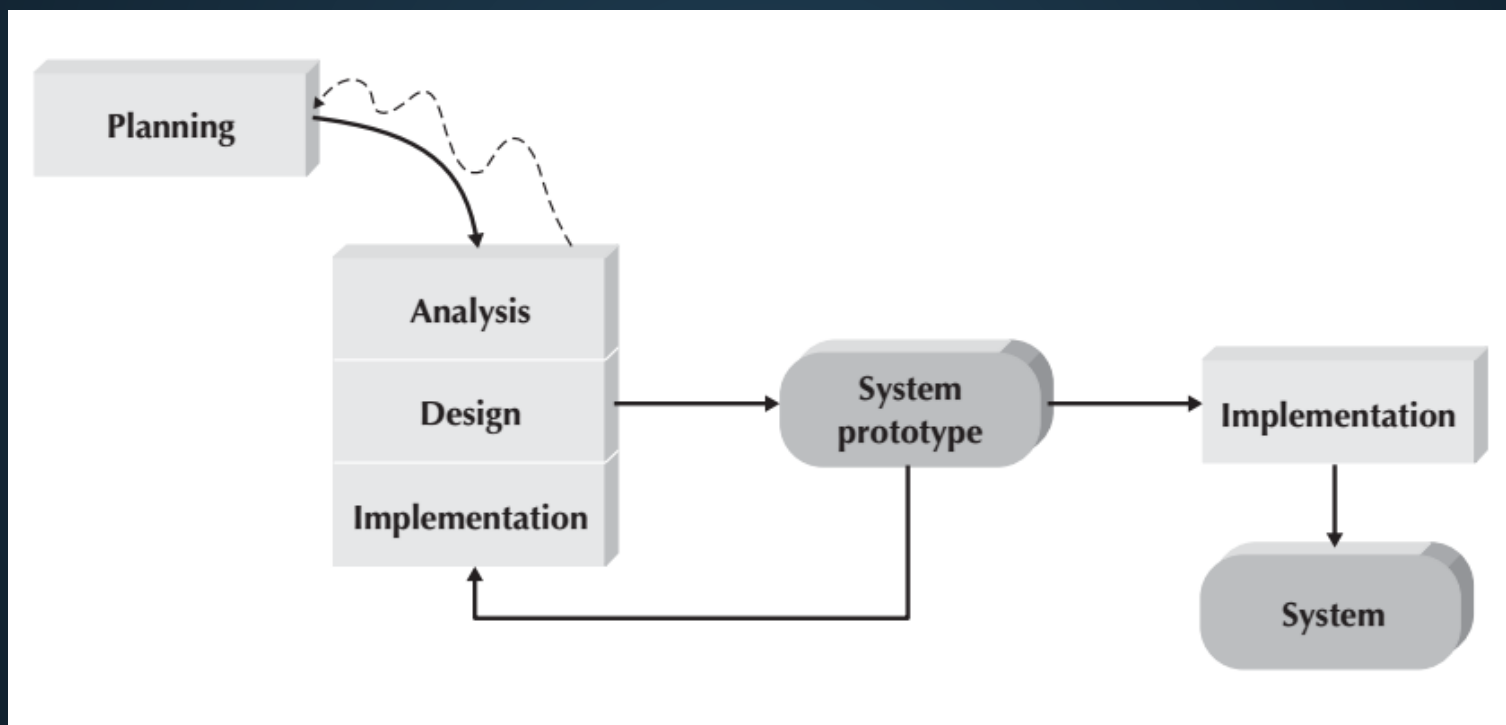


Prototyping(I)

- Performs the **analysis**, **design**, and **implementation** phases **concurrently**, and all three phases are performed **repeatedly in a cycle** until the system is completed.
- Basics of analysis and design are performed, and work immediately begins on a *system **prototype***, a **quick-and-dirty** program that provides a **minimal amount of features**. The first prototype is usually the first part of the system that is used.
- This is shown to the **users** and the **project sponsor**, who provide comments. These comments are used to **reanalyze, redesign, and re-implement** a **second** prototype, which provides a **few more features**. This process continues in a cycle until the analysts, users, and sponsor **agree** that the prototype provides enough functionality **to be installed** and used in the organization.
- After the prototype (now called the "**system**") is installed, refinement occurs until it is accepted as the new system.



Prototyping(II)





Prototyping(II)

مزیت اصلی این است که بسیار سریع سیستمی را فراهم می کند که کاربران می توانند با آن تعامل داشته باشند، حتی اگر در ابتدا برای استفاده گسترده سازمانی آماده نباشد.

- نمونه سازی به کاربران اطمینان می دهد که تیم پروژه بر روی سیستم کار می کند (تاخیر طولانی وجود ندارد که کاربران پیشرفت کمی در آن مشاهده کنند)، و نمونه سازی به بهبود سریعتر نیازهای واقعی کمک می کند.

اغلب نمونه اولیه دستخوش تغییرات قابل توجهی می شود که بسیاری از تصمیمات اولیه طراحی ضعیف می شوند این میتواند باعث ایجاد مشکلاتی در توسعه سیستمهای پیچیده شود، زیرا مسائل و مشکلات اساسی تا زمانی که فرایند توسعه به خوبی انجام نشود، شناسایی نمیشوند.

- The **key advantage** is that it **very quickly** provides a system with which the users can **interact**, even **if it is not ready** for widespread organizational use at first.
- Prototyping reassures the users that the **project team is working on the system** (there are **no long delays** in which the users see little progress), and prototyping helps to **more quickly refine** real requirements.
- The **major problem** is that its **fast-paced system releases** challenge attempts to conduct **careful, methodical analysis**. Often the prototype undergoes such significant changes that many **initial design decisions** become **poor** ones. This can cause problems in the development of **complex** systems because **fundamental issues** and problems **are not recognized** until well into the development process.



Throwaway Prototyping(I)

هر یک از این مسائل با تجزیه و تحلیل، طراحی و ساخت یک نمونه اولیه طراحی بررسی می شود.

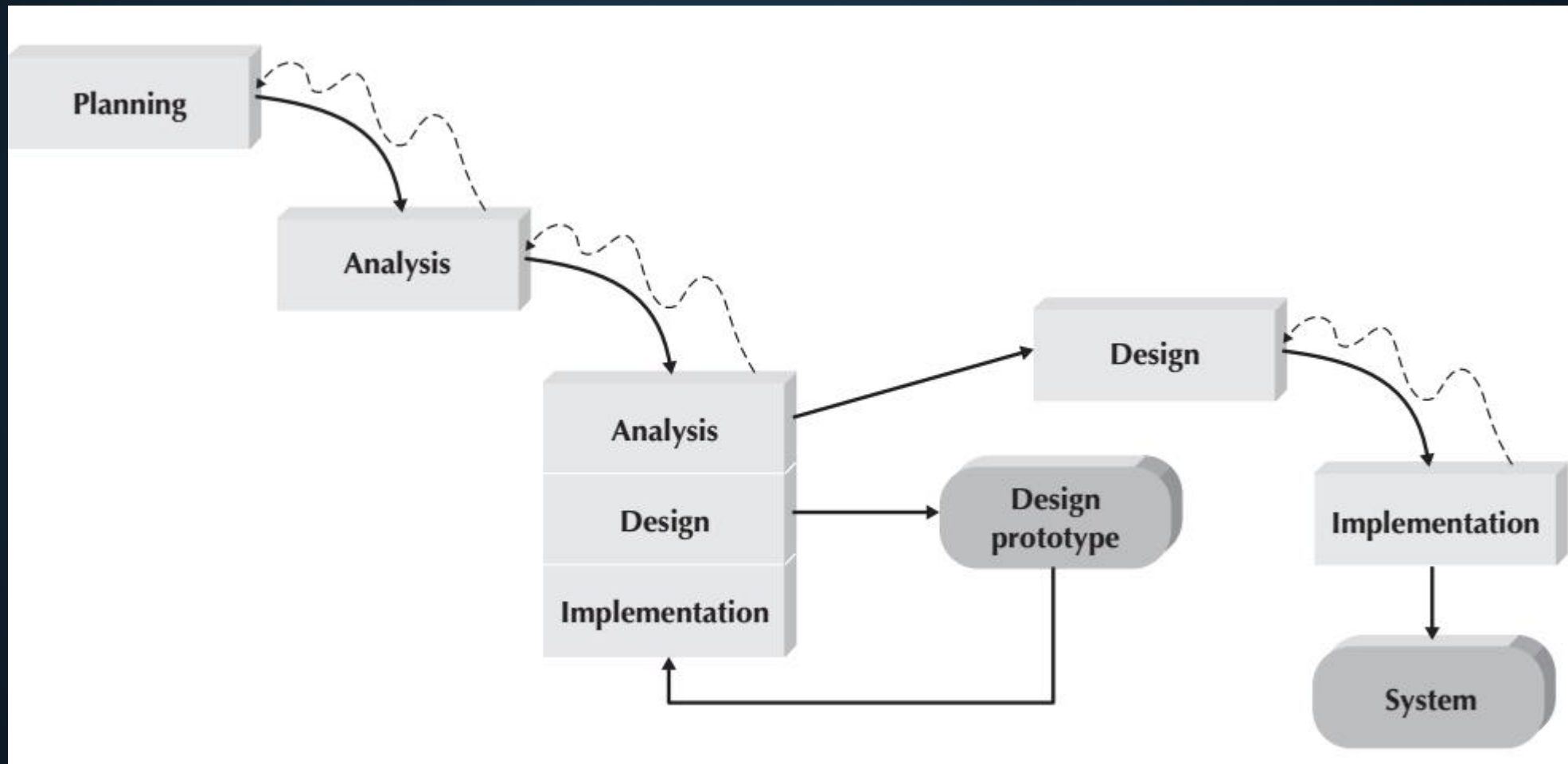
- These prototypes are used for a **very different purpose** than those previously discussed, and they have a **very different appearance**.
- It has a relatively thorough analysis phase that is used to gather information and to develop ideas for the system concept. However, users might **not** completely **understand** many of the **features** they suggest, and there may be challenging technical issues to be solved. Each of these **issues** is examined by **analyzing**, **designing**, and **building** a **design prototype**. A design prototype is not a working system; it is a **product that represents a part of the system** that needs additional refinement, and it contains only enough detail to **enable users to understand** the issues under consideration.



Throwaway Prototyping(II)

- Each of the prototypes is used to **minimize the risk** associated with the system by confirming that **important issues** are **understood** before the real system is built.
- Once the **issues are resolved**, the project moves into **design** and **implementation**. At this point, the **design prototypes are thrown away**, which is an important difference between these methodologies and prototyping methodologies, in which the prototypes evolve into the final system.
- It can **take longer to deliver** the final system as compared to prototyping-based methodologies, but produces **more stable and reliable systems**.

Throwaway Prototyping(III)





Agile

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

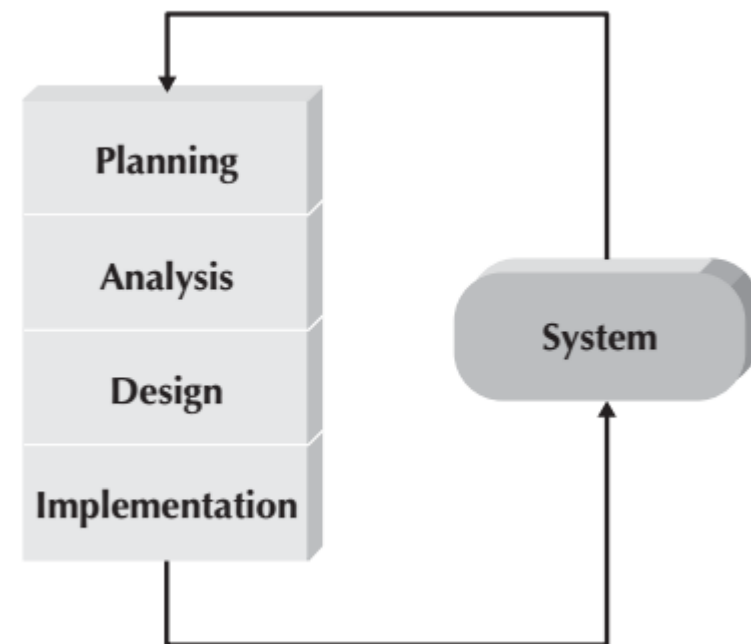
Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.





Selecting the Appropriate **Development Methodology**

Ability to Develop Systems	Structured Methodologies		RAD Methodologies		Agile Methodologies		
	Waterfall	Parallel	Phased	Prototyping	Throwaway Prototyping	XP	SCRUM
With Unclear User Requirements	Poor	Poor	Good	Excellent	Excellent	Excellent	Excellent
With Unfamiliar Technology	Poor	Poor	Good	Poor	Excellent	Good	Good
That Are Complex	Good	Good	Good	Poor	Excellent	Good	Good
That Are Reliable	Good	Good	Good	Poor	Excellent	Excellent	Excellent
With a Short Time Schedule	Poor	Good	Excellent	Excellent	Good	Excellent	Excellent
With Schedule Visibility	Poor	Poor	Excellent	Excellent	Good	Excellent	Excellent



Software Development Methodology(**SDM**)

چارچوبی برای به کارگیری شیوه های مهندسی نرم افزار با هدف
خاص ارائه ابزار لازم برای توسعه سیستم های فشرده نرم افزار

- A **framework** for applying software engineering practices with the specific aim of providing the necessary means for developing software-intensive systems.
- Have two parts.
 1. A set of **modeling conventions** comprising a Modeling **Language** (syntax and semantics)
 2. A **Process**, which
 - provides guidance as to the **order of the activities**,
 - specifies **what artifacts** should be developed using the Modeling Language,
 - **directs** the **tasks** of individual developers and the team as a whole,
 - offers **criteria** for monitoring and measuring a project's **products** and **activities**.

یک فرآیند، که

- در مورد ترتیب فعالیت ها راهنمایی می کند
- مشخص می کند که چه مصنوعات باید با استفاده از زبان مدل سازی توسعه داده شوند
- وظایف توسعه دهندگان فردی و تیم را به عنوان یک کل هدایت می کند
- معیارهایی برای نظارت و اندازه گیری محصولات و فعالیت های یک پروژه ارائه می دهد

مجموعه ای از قراردادهای مدل سازی شامل یک زبان مدل سازی
(نحو و معناشناسی)



Unified Modelling Language (UML)

هر توسعه دهنده متدولوژی و نشانه گذاری خاص خود را داشت

ارائه یک واژگان مشترک از اصطلاحات شی گرا و UML هدف تکنیک های نموداری به اندازه کافی غنی برای مدل سازی هر پروژه توسعه سیستمی از تجزیه و تحلیل تا پیاده سازی بود

- Each developer had his or her own methodology and notation.
- A **standard set of diagramming techniques**, *Unified Modeling Language(UML)*.
- The objective of UML was to provide a **common vocabulary** of **object-oriented terms** and **diagramming techniques** rich enough **to model any systems development project** from analysis through implementation.



References

- Dennis, Wixon, Tegarden, “System Analysis and Design, An Object Oriented Approach with UML”, 5th Edition, 2015.



What we will talk about next...

- Object-oriented principles
- RUP
- Scrum