

Software Engineering (1)

Lec 2

Software DEVELOPMENT LIFE CYCLE (SDLC)

دورة حياة تطوير البرمجيات (SDLC)

- Software development life cycle (SDLC) is an organizational process of developing and maintaining systems. It helps in establishing a system project plan, because it gives overall list of processes and sub- processes required for developing a system.

دورة حياة تطوير البرمجيات هي عملية تنظيمية لتطوير الأنظمة وصيانتها. يساعد في إنشاء خطة مشروع النظام ، لأنه يعطي قائمة شاملة من العمليات والعمليات الفرعية المطلوبة لتطوير النظام .

- A software life cycle model (also called process model) is a descriptive and diagrammatic representation of the software life cycle.

نموذج دورة حياة البرنامج (يسمى أيضًا نموذج العملية) هو تمثيل وصفي ورسمي لدورة حياة البرنامج

- A life cycle model represents all the activities required to make a software product transit through its life cycle phases.

يمثل نموذج دورة الحياة جميع الأنشطة المطلوبة لجعل منتج البرنامج يمر عبر مراحل دورة حياته .

- The System Development Life Cycle consists of four major phases and each may be composed of a series of steps.

تتكون دورة حياة تطوير النظام من أربع مراحل رئيسية وقد تتكون كل منها من سلسلة من الخطوات .

Software DEVELOPMENT LIFE CYCLE (SDLC)

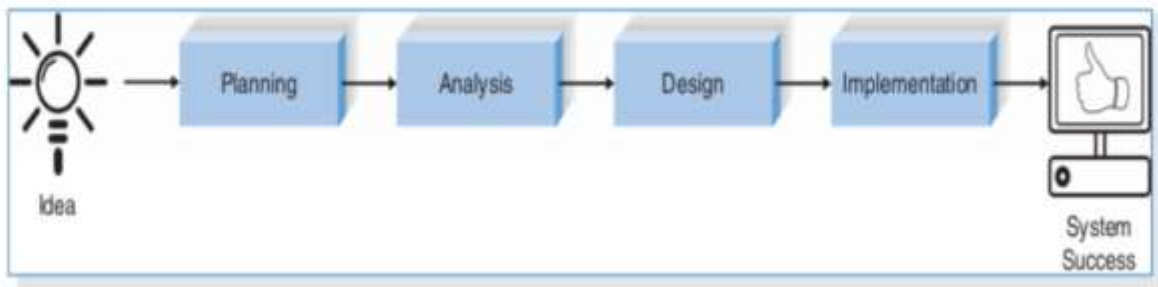


FIGURE 1-2
The Systems Development Life Cycle

Project Methodology Options :

خيارات منهجية المشروع

- the Systems Development Life Cycle (SDLC) provides the foundation for the processes used to develop an information system.
توفر دورة حياة تطوير الأنظمة الأساس للعمليات المستخدمة لتطوير نظام المعلومات .
- A methodology is a formalized approach to implementing the SDLC (i.e., it is a list of steps and deliverables).
المنهجية هي نهج رسمي لتنفيذ (أي أنها قائمة بالخطوات والنواتج) .
- There are many different systems development methodologies, and they vary in terms of the progression that is followed through the phases of the SDLC.
هناك العديد من منهجيات تطوير الأنظمة المختلفة ، وهي تختلف من حيث التقدم الذي يتم اتباعه خلال مراحل .

1. Classical Waterfall Model

نموذج الشلال الكلاسيكي

- With waterfall development, analysts and users proceed sequentially from one phase to the next.
مع تطوير الشلال ، ينتقل المحللون والمستخدمون بالتتابع من مرحلة إلى أخرى
- The key deliverables for each phase are typically voluminous (often, hundreds of pages) and are presented to the approval committee and project sponsor for approval as the project moves from phase to phase.
عادةً ما تكون المخرجات الرئيسية لكل مرحلة ضخمة (غالبًا ، مئات الصفحات) ويتم تقديمها إلى لجنة الموافقة وراعي المشروع للموافقة عليها مع انتقال المشروع من مرحلة إلى أخرى
- Once the work produced in one phase is approved, the phase ends and the next phase begins.
بمجرد الموافقة على العمل المنتج في مرحلة واحدة ، تنتهي المرحلة وتبدأ المرحلة التالية
- As the project progresses from phase to phase, it moves forward in the same manner as a waterfall.
مع تقدم المشروع من مرحلة إلى أخرى ، فإنه يتحرك للأمام بنفس طريقة الشلال (الكلاسيكال موديل ده بيعتمد ان كل مرحله بتخلص لوحدها مفيش تعديل بيحصل غير جوا المرحلة دي مينفعش يجي العميل يقول بعد م نخلص اي مرحله يجي يقول اصل انا افكرت اني عاوز اضيف حاجه فالمرحلة اللي قبلها مينفعش ولذلك سميت بالشلال عشان صعب جد يرجع لورا الطريقه دي رغم انها دقيقه الا انها فيها عيوب صعب التعامل معاها)

- While it is possible to go backward through the phases (e.g., from design back to analysis), it is quite difficult. (Imagine yourself as a salmon trying to swim upstream in a waterfall).

في حين أنه من الممكن العودة إلى الوراء خلال المراحل (على سبيل المثال ، من التصميم إلى التحليل) ، إلا أنه صعب للغاية. (تخيل نفسك كسمك السلمون يحاول السباحة عكس التيار في شلال). (يعني فالمودل ده انا مقدرش ارجع خطوه لورا)

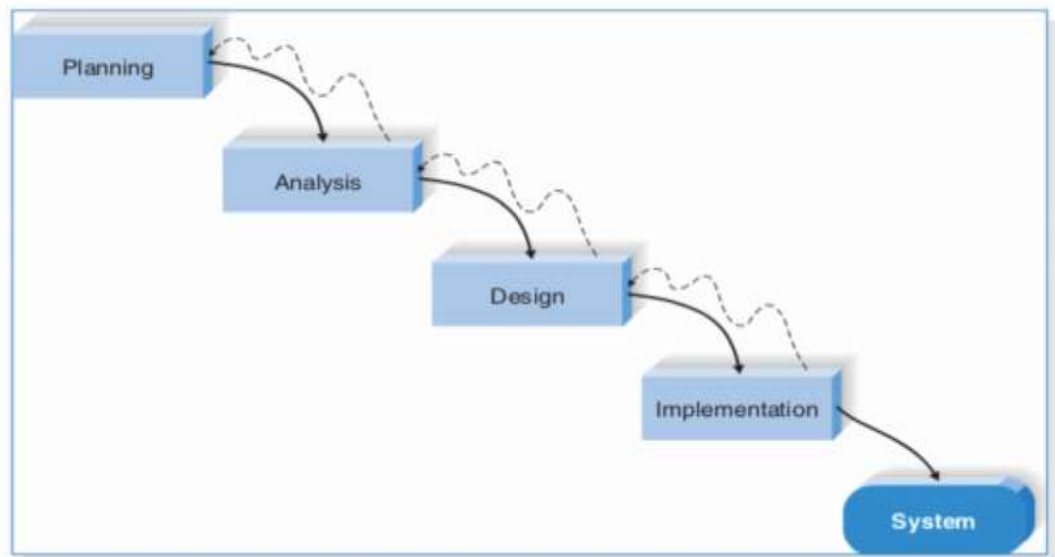


FIGURE 2-2
Waterfall Development

Advantage

المميزات

- Advantage: the advantages are: الميزة: المزايا هي
- identifying requirements long before programming begins, and تحديد المتطلبات قبل وقت طويل من بدء البرمجة ،
- limiting changes to the requirements as the project proceeds. الحد من التغييرات على المتطلبات أثناء تقدم المشروع .

Disadvantage: العيب

The key disadvantages are:

العيوب الرئيسية هي

- the design must be completely specified before programming begins, يجب تحديد التصميم بالكامل قبل بدء البرمجة (نظرا لان مينفعش ترجع مرحله لورا فمش بقدر اغير في الديزاين بتاعها بعد م اخلص)

- a long time elapses between the completion of the system proposal in the analysis phase and the delivery of system,

ينقضي وقت طويل بين الانتهاء من اقتراح النظام في مرحلة التحليل وتسليم النظام (بتستغرق وقت طويل جدا عشان تكمل كل مرحله وتوصل للكمال فكل واحد وع قد ماهو ميزه ع قد ماهو عيب لان الوقت اللي بناخده مش متناسب مع رؤيه السوق حاليا لان كل يوم بيحصل تغيير فمتطلبات البيزنس ولان مش هقدر اسمح للعميل ان لو حصل اي تغييرات مش هقدر اضيفهاله وكمان ممكن الديزاين من م اوصل يكون مش مناسب)

- testing is treated almost as an afterthought in the implementation phase,

يتم التعامل مع الاختبار كفكرة لاحقة تقريبًا في مرحلة التنفيذ ،

- poor communication mechanism, so important requirements may be overlooked in the volumes of documentation. If the project team misses an important requirement, expensive post-implementation programming may be needed,

آلية اتصال ضعيفة ، لذلك قد يتم التغاضي عن المتطلبات المهمة في مجلدات الوثائق. إذا فات فريق المشروع شرطًا مهمًا ، فقد تكون هناك حاجة لبرمجة ما بعد التنفيذ باهظة الثمن ،

- Users may forget the original purpose of the system, since so much time has elapsed between the original idea and actual implementation.

قد ينسى المستخدمون الغرض الأصلي من النظام ، حيث انقضى وقت طويل بين الفكرة الأصلية والتنفيذ الفعلي

- in today's dynamic business environment, a system that met the existing environmental conditions during the analysis phase may need considerable rework to match the environment when it is implemented. This rework requires going back to the initial phase and making needed changes through each of the subsequent phases in turn.

في بيئة الأعمال الديناميكية الحالية ، قد يحتاج النظام الذي يلبي الظروف البيئية الحالية أثناء مرحلة التحليل إلى إعادة صياغة كبيرة لمطابقة البيئة عند تنفيذه. تتطلب إعادة العمل هذه العودة

إلى المرحلة الأولية وإجراء التغييرات اللازمة خلال كل مرحلة من المراحل اللاحقة على التوالي .

When the Waterfall SDLC Model can be Used?!

متى يمكن استخدام نموذج Waterfall SDLC ؟

- It is not a practical model in the sense that it cannot be used in actual software development projects.

إنه ليس نموذجًا عمليًا بمعنى أنه لا يمكن استخدامه في مشاريع تطوير البرامج الفعلية .

- Thus, this model can be considered to be a theoretical way of developing software.

وبالتالي ، يمكن اعتبار هذا النموذج طريقة نظرية لتطوير البرمجيات .

- But all other life cycle models are essentially derived from the classical waterfall model.

لكن جميع نماذج دورة الحياة الأخرى مشتقة أساسًا من نموذج الشلال الكلاسيكي

- So, in order to be able to appreciate other life cycle models it is necessary to learn the classical waterfall model.

لذلك ، لكي تكون قادرًا على تقدير نماذج دورة الحياة الأخرى ، من الضروري تعلم نموذج الشلال الكلاسيكي .

2. Parallel Development Methodology منهجية التنمية الموازية

(بدأ مهندسي البرمجيات يفكرون ان النموذج الكلاسيكي مش عملي لان كل ما البروجيكت يكبر المشكله بتكبر فقررنا يطورو النموذج الكلاسيكي ده عشان يقدرنا يرجعوا خطوه لورا ويقلل كل الوقت الطويل اللي بيتم استغلاله فكل مرحله ده فعملوا اول تطوير للكلاسيكي والهدف منه تقليل الوقت اللي بيتم استغلاله فكل مرحله وتقليل حجم المشكله كل م البروجيكت يزيد .

- one of the waterfall variations is the parallel development methodology.

أحد أشكال الشلال هو منهجية التطوير الموازية .

- It evolved to address the lengthy time frame of waterfall development.
تطورت لمعالجة الإطار الزمني الطويل لتطوير الشلال .

- instead of doing the design and implementation in sequence, a general design for the whole system is performed.

بدلاً من القيام بالتصميم والتنفيذ بالتسلسل ، يتم تنفيذ تصميم عام للنظام بأكمله .

(هناقل الوقت فمرحلة التصميم لما نقسم البروجيكت لمجموعات تاسكات وبتوزيع أعضاء الفريق وكل جروب بياخذ تاسك معين وبيشتغلوا مع بعض وكل جروب بيطلع جزء من البروجيكت وبعد كذا بجمع كل اللي الاجزاء اللي وصلناها وبالتالي قللت الوقت اللي بناخده فالنموذج الكلاسيكال وكذا هنقدر نواكب تغييرات السوق)

- Then the project is divided into a series of subprojects that can be designed and implemented in parallel.

ثم يتم تقسيم المشروع إلى سلسلة من المشاريع الفرعية التي يمكن تصميمها وتنفيذها بالتوازي .

- Once all subprojects are completed, there is a final integration of the separate pieces, and the system is delivered.

بمجرد الانتهاء من جميع المشاريع الفرعية ، يكون هناك تكامل نهائي للأجزاء المنفصلة ، ويتم تسليم النظام .

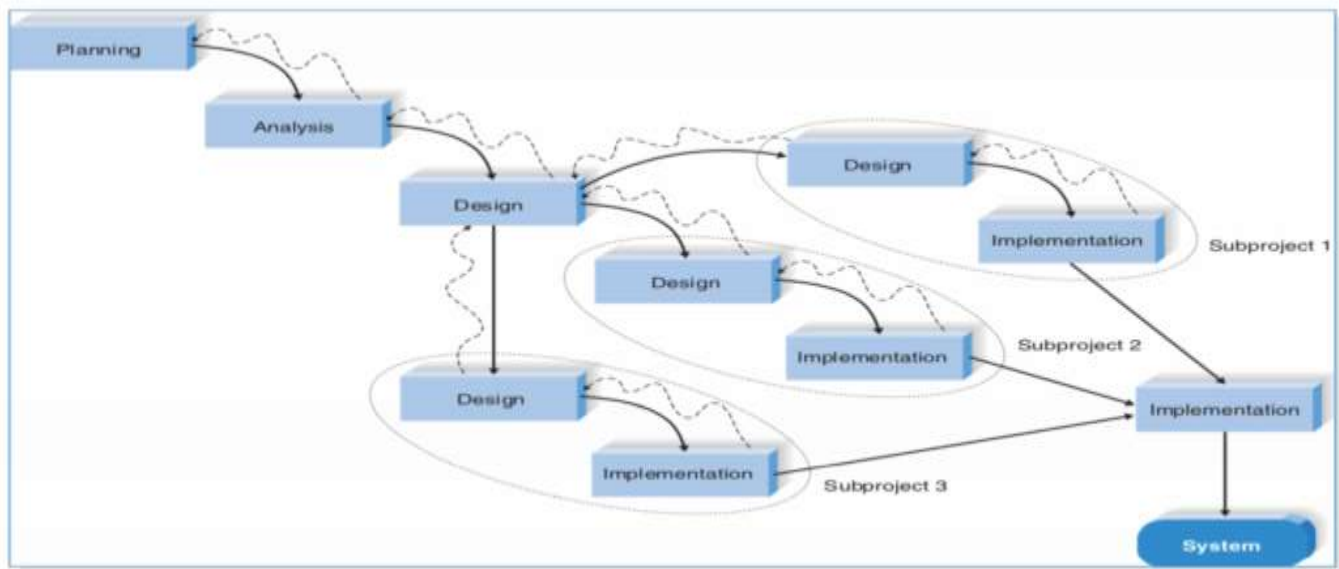


FIGURE 2-3
Parallel Development

Advantage مميزات

- reduces the time required to deliver a system,
يقلل من الوقت المطلوب لتقديم النظام ،
- So, changes in the business environment are less likely to produce the need for rework. Disadvantage

لذلك ، فإن التغييرات في بيئة الأعمال أقل احتمالاً لإنتاج الحاجة إلى إعادة العمل

Disadvantage عيوب

- suffers from problems caused by voluminous deliverables.
يعاني من مشاكل ناجمة عن المخرجات الضخمة .
- It also adds a new problem: If the subprojects are not completely independent, design decisions in one subproject may affect another, and at the project end, integrating the subprojects may be quite challenging.

كما أنه يضيف مشكلة جديدة: إذا لم تكن المشاريع الفرعية مستقلة تمامًا ، فقد تؤثر قرارات التصميم في مشروع فرعي واحد على مشروع فرعي آخر ، وفي نهاية المشروع ، قد يكون تكامل المشاريع الفرعية أمرًا صعبًا للغاية .

When the parallel development methodology can be used?!

متى يمكن استخدام منهجية التطوير الموازي

- For the projects where an accurate product or service production can be separated into independent tasks.
بالنسبة للمشاريع التي يمكن فيها تقسيم إنتاج منتج أو خدمة دقيق إلى مهام مستقلة .
- For the small and mid-sized projects, where requirements are strictly predefined

للمشاريع الصغيرة والمتوسطة الحجم ، حيث يتم تحديد المتطلبات مسبقًا بدقة

- The software engineers are **qualified enough to separate and integrate** the subprojects effectively

مهندسو البرمجيات مؤهلون بما يكفي لفصل المشاريع الفرعية ودمجها بفعالية

3. Iterative Model

النموذج التكراري

- breaks the overall project into a series of versions that are developed sequentially.

يقسم المشروع ككل إلى سلسلة من الإصدارات التي يتم تطويرها بالتتابع .

- The most important and fundamental requirements are bundled into the first version of the system.

تم تجميع المتطلبات الأساسية والأكثر أهمية في الإصدار الأول من النظام .

- This version is developed quickly by a mini-waterfall process, and once implemented;

تم تطوير هذا الإصدار بسرعة من خلال عملية شلال صغير ، وبمجرد تنفيذه ؛

- the users can provide valuable feedback to be incorporated into the next version of the system.

يمكن للمستخدمين تقديم ملاحظات قيمة ليتم دمجها في الإصدار التالي من النظام .

(بدأو يطورو عشان يتلافوا الاخطاء اللي فانت فعلو الموديل ده فبدأ النموذج ده يطلع مجموعه من النسخ كل نسخه فيهم بتمثل مجموعه من المتطلبات بس مهندسي البرمجيات عملوا حاجه مهمه جدا وهي انهم قسموا المتطلبات اللي جمعها حسب اهميتها ويشوف اكثر متطلبات مهمه وشايفها اساسيه ويطلعها فأول نسخه عشان يكسب ثقه العميل ومن خلالها العميل هيشغل بإيده وشاف البروجيكت بشكل مبدئي ويجربه ولو حب يعدل علي حاجه يقدر يعدلها ويضيف المتطلبات اللي محتاجها العميل ويطلع النسخه الثانيه ونعرضه عليه ونفس الكلام ويفضل يعدل عليه لحد م نوصل للنسخ النهائي) .

من مرحله Analysis بقدر اعدل علي البروجيكت واغير فيه اللي انا عاوزه .

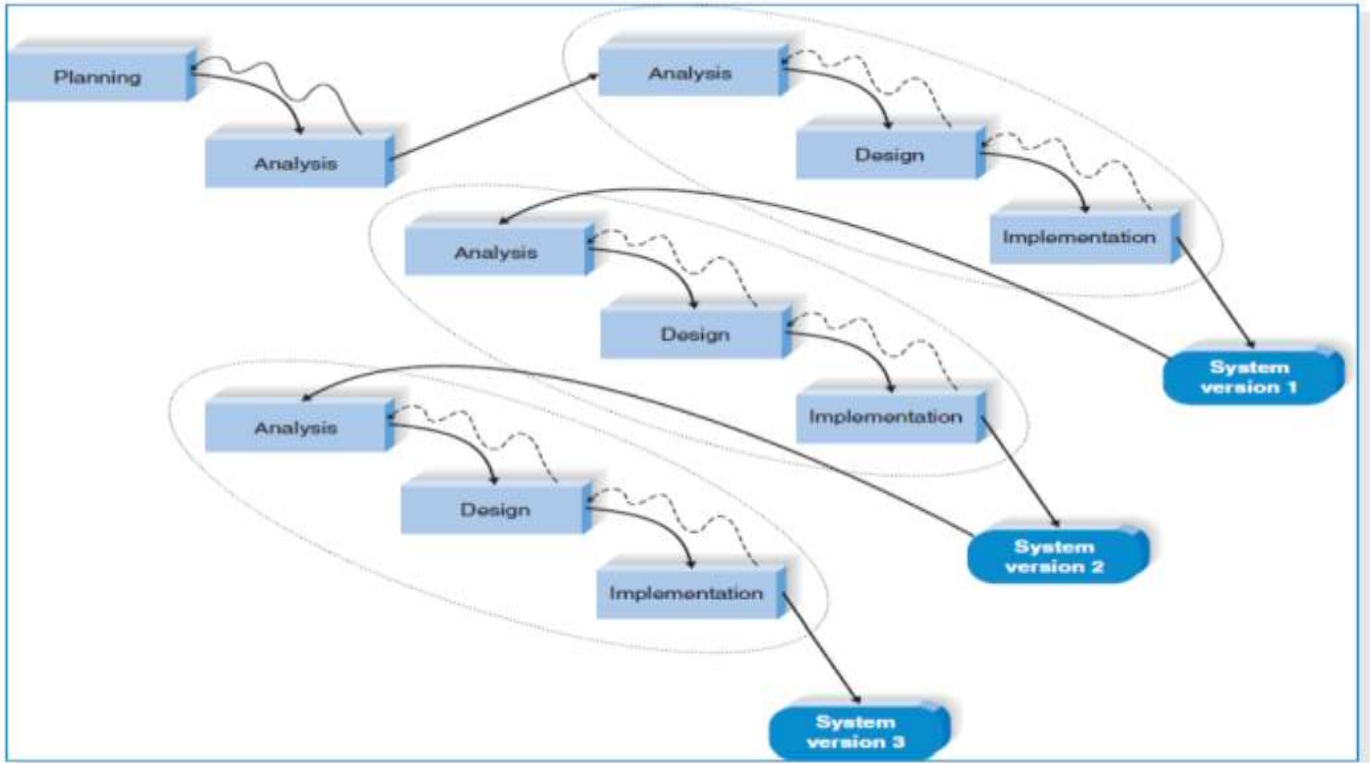


FIGURE 2-5
Iterative Development

• **Advantage:**

مميزات

- Iterative development gets a preliminary version of the system to the users quickly so that business value is provided.

يُحصل التطوير المتكرر على نسخة أولية من النظام للمستخدمين بسرعة بحيث يتم توفير قيمة الأعمال .

- Since users are working with the system, important additional requirements may be identified and incorporated into subsequent versions.

نظرًا لأن المستخدمين يعملون مع النظام ، فقد يتم تحديد متطلبات إضافية مهمة ودمجها في الإصدارات اللاحقة .

(في الموديل ده العميل بيكون جزء من المنظومه بتاعتي فلازم يكون عندي ذكاء واعرف اوصل للعميل متطلباته اللي هو عاوزها بالظبط)

• Disadvantage:

العيوب

- users begin to work with a system that is intentionally incomplete. Users must accept that only the most critical requirements of the system will be available in the early versions and must be patient with the repeated introduction of new system versions.

يبدأ المستخدمون في العمل بنظام غير مكتمل عن قصد. يجب أن يقبل المستخدمون أن أكثر متطلبات النظام أهمية هي التي ستكون متاحة في الإصدارات المبكرة ويجب عليهم التحلي بالصبر مع التقديم المتكرر لإصدارات النظام الجديدة .

When the Iterative Development methodology can be used?!

متى يمكن استخدام منهجية التطوير التكراري

- The requirements to the final product are carefully predefined.
متطلبات المنتج النهائي محددة مسبقًا بعناية
- Applied to the large-scale projects.
تطبق على المشاريع الكبيرة .
- The main task is predefined, but the details may advance with the time
المهمة الرئيسية محددة مسبقًا ، لكن التفاصيل قد تتقدم مع مرور الوقت

Throwaway Prototyping

- throwaway prototyping has a fairly **deep analysis phase** that is used to :
للنماذج الأولية مرحلة تحليل عميقة إلى حد ما تُستخدم من أجل :
- gather requirements and
جمع المتطلبات و
- develop ideas for the system concept.
تطوير أفكار لمفهوم النظام

- Many of the features suggested by the users may not be well understood, and there may be challenging technical issues to be solved.

قد لا تكون العديد من الميزات التي اقترحها المستخدمون مفهومة جيداً ، وقد تكون هناك مشكلات فنية صعبة يجب حلها

- Each of these issues is examined by analyzing, designing, and building a design prototype.

يتم فحص كل من هذه القضايا من خلال تحليل وتصميم وبناء نموذج أولي للتصميم

- A design prototype is not intended to be a working system.

لا يُقصد من النموذج الأولي للتصميم أن يكون نظاماً عاملاً

- It contains only enough detail to enable users to understand the issues under consideration.

يحتوي فقط على تفاصيل كافية لتمكين المستخدمين من فهم القضايا قيد النظر .

- For example, suppose that users are not completely clear on how an order entry system should work. The analyst team might build a series of HTML pages to be viewed on a Web browser to help the users visualize such a system.

على سبيل المثال ، افترض أن المستخدمين ليسوا واضحين تماماً بشأن كيفية عمل نظام إدخال الأمر. قد يقوم فريق المحلل ببناء سلسلة من الصفحات ليتم عرضها على مستعرض ويب لمساعدة المستخدمين على تصور مثل هذا النظام .

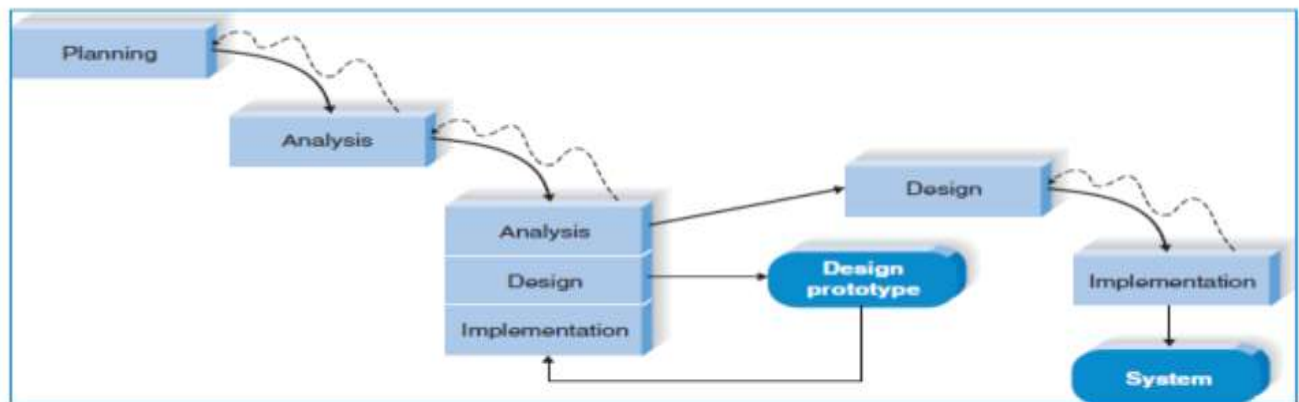


FIGURE 2-7
Throwaway Prototyping

- 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.

تُستخدم النماذج الأولية لتقليل المخاطر المرتبطة بالنظام من خلال التأكيد على فهم القضايا المهمة قبل بناء النظام الحقيقي .

- Once the issues are resolved, the project moves into design and implementation.

بمجرد حل المشكلات ، ينتقل المشروع إلى التصميم والتنفيذ

- the prototypes evolve into the final system.

النماذج الأولية تتطور إلى النظام النهائي

- **Advantage:** مميزات

refine key issues before a system is built.

صقل القضايا الرئيسية قبل بناء النظام

- **Disadvantage:** عيوب

It may take longer to deliver the final system, but the approach usually produces more stable and reliable systems.

قد يستغرق تسليم النظام النهائي وقتًا أطول ، لكن هذا النهج ينتج عادةً أنظمة أكثر استقرارًا وموثوقية .

When the Throwaway prototyping methodology can be used?!

متى يمكن استخدام منهجية النماذج الأولية للتخلص منها

- Complex systems أنظمة معقدة
- To produce more stable and reliable systems

لإنتاج أنظمة أكثر استقرارًا وموثوقية

Agile Development

تطوير البرامج بتقنية أجيل

- Agile development is a group of programming-centric methodologies that focus on streamlining the SDLC.

التطوير السريع هو مجموعة من المنهجيات التي تتمحور حول البرمجة والتي تركز على تبسيط

- Much of the modeling and documentation overhead is eliminated; instead, face-to-face communication is preferred.

يتم التخلص من الكثير من النفقات العامة للنمذجة والتوثيق ؛ بدلاً من ذلك ، يفضل التواصل وجهاً لوجه .

- A project emphasizes simple, iterative application development in which every iteration is a complete software project, including planning, requirements analysis, design, coding, testing, and documentation.

يركز المشروع على تطوير التطبيقات البسيط والتكراري حيث يكون كل تكرار عبارة عن مشروع برمجي كامل ، بما في ذلك التخطيط وتحليل المتطلبات والتصميم والترميز والاختبار والتوثيق .



(بنسخدمها لما تكون المتطلبات بتتغير كل شويه وبتكون موفره فالمشروعات الكبيره)

Agile Development

تطوير البرامج بتقنية أجيل

- At the end of each sprint, a potentially shippable product increment is delivered.

في نهاية كل سباق ، يتم تسليم زيادة محتملة في المنتج قابلة للشحن .

- Thus, with every iteration new features are added to the product, which results in the gradual project growth.

وبالتالي ، مع كل تكرار ، تتم إضافة ميزات جديدة إلى المنتج ، مما يؤدي إلى نمو المشروع التدريجي

- With the features being validated so early in the development, the chances of delivering a potentially failed product are significantly lower.

مع التحقق من صحة الميزات في وقت مبكر جدًا من التطوير ، تكون فرص تقديم منتج محتمل الفشل أقل بكثير .

When the Agile methodology can be used?!

متى يمكن استخدام منهجية Agile؟

- The users' needs change dynamically .

تتغير احتياجات المستخدمين ديناميكيًا

- Less price for the changes implemented because of the many iterations

سعر أقل للتغييرات التي تم تنفيذها بسبب التكرارات العديدة

- Unlike the Waterfall model, it requires only initial planning to start the project

بخلاف نموذج الشلال ، لا يتطلب الأمر سوى التخطيط الأولي لبدء المشروع

Selecting the Appropriate Development Methodology The important methodology selection criteria are listed:

اختيار منهجية التطوير المناسبة يتم سرد معايير اختيار المنهجية الهامة

Clarity of User Requirements:

وضوح متطلبات المستخدم

- When the user requirements for what the system should do are unclear,

عندما تكون متطلبات المستخدم لما يجب أن يفعله النظام غير واضحة ،

- it is difficult to understand them by talking about them and explaining them with written reports.

من الصعب فهمها بالحديث عنها وشرحها بتقارير مكتوبة .

- Users need to interact with technology to really understand what the new system can do and how to best apply it to their needs.

يحتاج المستخدمون إلى التفاعل مع التكنولوجيا ليفهموا حقًا ما يمكن أن يفعله النظام الجديد وكيفية تطبيقه على أفضل وجه لاحتياجاتهم

- throwaway prototyping is usually more appropriate when user requirements are unclear,. Agile development may also be appropriate if on-site user input is available.

عادة ما تكون النماذج الأولية أكثر ملاءمة عندما تكون متطلبات المستخدم غير واضحة. قد يكون التطوير السريع مناسبًا أيضًا في حالة توفر مدخلات المستخدم في الموقع

The important methodology selection criteria are listed:

يتم سرد معايير اختيار المنهجية الهامة

وضوح متطلبات المستخدم
Clarity of User Requirements:

- When the user requirements for what the system should do are unclear,

عندما تكون متطلبات المستخدم لما يجب أن يفعله النظام غير واضحة ،

- it is difficult to understand them by talking about them and explaining them with written reports.

من الصعب فهمها بالحديث عنها وشرحها بتقارير مكتوبة .

- Users need to interact with technology to really understand what the new system can do and how to best apply it to their needs.

يحتاج المستخدمون إلى التفاعل مع التكنولوجيا ليفهموا حقًا ما يمكن أن يفعله النظام الجديد وكيفية تطبيقه على أفضل وجه لاحتياجاتهم

- throwaway prototyping is usually more appropriate when user requirements are unclear,. Agile development may also be appropriate if on-site user input is available.

عادة ما تكون النماذج الأولية أكثر ملاءمة عندما تكون متطلبات المستخدم غير واضحة. قد يكون التطوير السريع مناسبًا أيضًا في حالة توفر مدخلات المستخدم في الموقع.

Familiarity with Technology: الإلمام بالتكنولوجيا

- When the system will use new technology which the analysts and programmers are not familiar,

عندما يستخدم النظام تقنية جديدة لا يعرفها المحللون والمبرمجون ،

- applying the new technology early in the methodology will improve the chance of success.

سيؤدي تطبيق التكنولوجيا الجديدة في وقت مبكر من المنهجية إلى تحسين فرصة النجاح

- **Throwaway prototyping** is particularly appropriate for situations where there is a lack of familiarity with technology, because it explicitly encourages the developers to create design prototypes for areas with high risks.

تعد النماذج الأولية ذات الصلة مناسبة بشكل خاص للحالات التي يكون فيها نقص في الإلمام بالتكنولوجيا ، لأنها تشجع المطورين بشكل صريح على إنشاء نماذج أولية للتصميم للمناطق ذات المخاطر العالية

- **Iterative development** is good as well, because opportunities are created to investigate the technology in some depth before the design is complete.

يعد التطوير التكراري جيدًا أيضًا ، لأنه يتم إنشاء الفرص للتحقيق في التكنولوجيا بشيء من العمق قبل اكتمال التصميم .

System Complexity: تعقيد النظام

- Complex systems require careful and detailed analysis and design.
تتطلب الأنظمة المعقدة تحليلًا وتصميمًا دقيقًا وتفصيليًا .

- **Throwaway prototyping** is particularly well suited to such detailed analysis and design,
النماذج الأولية التي يتم التخلص منها مناسبة تمامًا لمثل هذا التحليل والتصميم التفصيليين ،

- **The waterfall methodologies** can handle complex systems, but without the ability to get the system or prototypes into users' hands early on, some key issues may be overlooked.

يمكن لمنهجيات الشلال التعامل مع الأنظمة المعقدة ، ولكن بدون القدرة على وضع النظام أو النماذج الأولية في أيدي المستخدمين في وقت مبكر ، قد يتم التغاضي عن بعض المشكلات الرئيسية

- Although **iterative development** methodologies enable users to interact with the system early in the process, we have observed that project teams who follow these methodologies tend to devote less attention to the analysis of the complete problem domain than they might if they were using other methodologies.

على الرغم من أن منهجيات التطوير التكراري تمكن المستخدمين من التفاعل مع النظام في وقت مبكر من العملية ، فقد لاحظنا أن فرق المشروع التي تتبع هذه المنهجيات تميل إلى تكريس اهتمام أقل لتحليل مجال المشكلة بالكامل مما قد يحدث إذا كانوا يستخدمون منهجيات أخرى .

Short Time :

وقت قصير

- Schedules Projects that have short time schedules are well suited for prototype and iterative methodologies because those methodologies are designed to increase the speed of development.

الجدول الزمنية تعتبر المشروعات ذات الجداول الزمنية القصيرة مناسبة تمامًا للنماذج الأولية والمنهجيات التكرارية لأن هذه المنهجيات مصممة لزيادة سرعة التطوير .

- excellent choices when time lines are short because they best enable the project team to adjust the functionality in the system on the basis of a specific delivery date.

اختيارات ممتازة عندما تكون الخطوط الزمنية قصيرة لأنها تمكن فريق المشروع على أفضل وجه من ضبط الوظائف في النظام على أساس تاريخ التسليم المحدد.

- If the project schedule starts to slip, it can be readjusted by removal of the functionality from the version or prototype under development.

إذا بدأ الجدول الزمني للمشروع في الانزلاق ، فيمكن إعادة ضبطه عن طريق إزالة الوظيفة من الإصدار أو النموذج الأولي قيد التطوير .

- Waterfall-based methodologies are **the worst choice** when time is at a premium, because they do not allow for easy schedule changes.

المنهجيات القائمة على الشلال هي الخيار الأسوأ عندما يكون الوقت مرتفعًا ، لأنها لا تسمح بإجراء تغييرات سهلة على الجدول الزمني .

Schedule Visibility:

الجدول الزمني الرؤية

- One of the greatest challenges in systems development is knowing whether a project is on schedule.

أحد أكبر التحديات في تطوير الأنظمة هو معرفة ما إذا كان المشروع في الموعد المحدد

- This is particularly true of the waterfall-based methodologies because design and implementation occur at the end of the project.

هذا صحيح بشكل خاص للمنهجيات القائمة على الشلال لأن التصميم والتنفيذ يحدثان في نهاية المشروع .

- The prototype and iterative methodologies move many of the critical design decisions to a position earlier in the project to help project managers recognize and address risk factors and keep expectations in check.

ينقل النموذج الأولي والمنهجيات التكرارية العديد من قرارات التصميم الحاسمة إلى موقع سابق في المشروع لمساعدة مديري المشاريع على التعرف على عوامل الخطر ومعالجتها والحفاظ على التوقعات قيد الفحص .

Question

1. We can use For the small and mid-sized projects.

(waterfall – parallel – iterative)

2. We can use When requirements are unclear

(Waterfall – iterative, prototyping, Agile – Parallel)

بستخدمهم لما يكون عندي المتطلبات مش واضحة فلازم اخلي العميل من ضمن المنظومه

3. We can use When the user nonfamiliar with technology.

(Waterfall – Prototyping , iterative – Parallel – Agile)

4. We can use When System is Complex

(Waterfall – Prototyping , iterative – Parallel – Agile)

5. We can use to minimize the time

(Waterfall – iterative, prototyping, Agile – Parallel)

بس Agile مكلفه اكثر فمش مناسبه اوي