แก้ไขครั้งล่าสุดเมื่อวันที่ 3 กุมภาพันธ์ 2568

2301361

SYSTEMS ANALYSIS AND DESIGN

2

Project Selection and Management

Most IT departments face a demand for IT projects that far exceeds the department’s ability to supply them. In the past 10 years, business application growth has exploded, and **chief information officers (CIOs)** are challenged to select projects that will provide the highest possible return on IT investments while managing project risk.

Historically, IT departments have tended to select projects by ad hoc methods: first-in, firstout;
pull, influence the wheel that squeaks the loudest, is the one that gets the grease
political clout; or the squeaky wheel getting the grease. In recent years, IT departments have collected project information and mapped the projects’ contributions to business goals, using a project portfolio perspective.

Project portfolio management, a process of selecting, prioritizing, and monitoring project results, has become a critical success factor for IT departments facing too many potential projects with too few resources. Software for project portfolio management, such as Hewlett Packard’s Project and Portfolio Management, Primavera Systems’ ProSight, and open-source Project.net, has become a valuable tool for IT organizations.

กรณีเป็นผู้รับเหมาก็ต้องทำคล้าย ๆ องค์กรคือ prioritize โครงการที่เสนอเข้ามา ว่าจะเซ็นสัญญาโครงการไหนก่อน โดยพิจารณาจากหลาย ๆ ปัจจัย เช่น ผลกำไร × ความน่าจะเป็นที่จะทำสำเร็จ ความน่าจะเป็นขึ้นกับความซับซ้อนของตัวงาน ความเชี่ยวชาญของผู้รับเหมา ทรัพยากรบุคคลและเวลา ลูกค้าคุยง่าย/ยาก เก็บเงินได้เร็ว/ช้า

Although training and software are available to help project managers, **unreasonable demands** set by project sponsors and business managers can make project management exceedingly difficult. Too often, the approach of the holiday season, the chance at winning a proposal with a low bid, or a funding opportunity pressures project managers to promise systems long before they are realistically able to deliver them. **These overly optimistic timetables are thought to be one of the biggest problems that projects face; instead of pushing a project forward faster, they result in delays.**

CONCEPTS IN ACTION 2-A Project Portfolio Management: An Essential Tool for IT Departments

Information systems are at the core of Sabre Holdings Corporation. The Sabre reservation system is the booking system of choice for travel agencies worldwide. Sabre is also the parent company of Travelocity.com, the second largest online travel agency in the United States.

Like many companies, Sabre's IT department struggles with many more project requests than it has resources to accomplish—as many as 1,500 proposals for 600 funded projects annually. Because of the volatile, competitive nature of the travel industry, Sabre is especially challenged to be certain that IT is doing the right projects under constantly changing conditions. While traditional project management techniques focus on getting individual projects done, Sabre needs to be able to rapidly change the entire set of projects it is working on as market conditions shift.

Project portfolio management software collects and manages information about all projects—those that are underway and those that are awaiting approval. The software helps prioritize projects, allocate employees, monitor projects in real time, flag cost and time variances, measure the ROI, and help the IT department objectively measure the efficiency and efficacy of IT investments.

Primavera Systems' PPM software has enabled Sabre Holdings to update its queue of projects regularly, and projects are now prioritized quarterly instead of annually. A study of users of Hewlett Packard's PPM Center software found that in all cases, the investment in the software paid for itself in a year. Other findings were an average 30% increase in on-time projects, a 12% reduction in budget variance, and a 30% reduction in the amount of time IT spent on project reporting.

Sources: Tucci, Linda, "Project portfolio management takes flight at Sabre," SearchCIO.com, November 28, 2007.
Tucci, Linda, "PPM strategy a CIO's must-have in hard times," SearchCIO.com, March 5, 2008.

Size	What is the size? How many people are needed to work on the project?
Cost	How much will the project cost the organization?
Purpose	What is the purpose of the project? Is it meant to improve the technical infrastructure? Support a current business strategy? Improve operations? Demonstrate a new innovation?
Length	How long will the project take before completion? How much time will go by before value is delivered to the business?
Risk	How likely is it that the project will succeed or fail?
Scope	How much of the organization is affected by the system? A department? A division? The entire corporation?
Economic Value	How much money does the organization expect to receive in return for the amount the project costs?

CONCEPTS IN ACTION 2-B Interview with Lyn McDermid, CIO, Dominion Virginia Power

A CIO needs to have a global view when identifying and selecting projects for her organization. I would get lost in the trees if I were to manage on a project-by-project basis. Given this, I categorize my projects according to my three roles as a CIO, and the mix of my project portfolio changes depending on the current business environment.

My primary role is to **keep the business running**. That means every day when each person comes to work, they can perform his or her job efficiently. I measure this using various service levels, cost, and productivity measures. Projects that keep the business running could have a high priority if the business were in the middle of a merger, or a low priority if things were running smoothly, and it were "business as usual."

My second role is to push **innovation that creates value for the business**. I manage this by looking at our lines of business and asking which lines of business create the most value for the company. These are the areas for which I should be providing the most value. For example, if we had a highly innovative marketing strategy, I would push for innovation there. If operations were running smoothly, I would push less for innovation in that area.

My third role is strategic, to look beyond today and find **new opportunities** for both IT and the business of providing energy. This may include investigating process systems, such as automated meter reading or looking into the possibilities of wireless technologies. *Lyn McDermid*

ยกตัวอย่างบริษัท Google

สำคัญที่ 1 ทำระบบที่สนับสนุนงานหลัก/งานประจำขององค์กรก่อน เช่น Google Search

สำคัญที่ 2 ทำระบบที่ช่วยเพิ่มคุณค่า (value) ให้องค์กร เช่น Gmail, Google Drive, Google Cloud Platform (สร้างระบบนิเวศ ecological system เช่น ระบบนิเวศของ Apple, Android, Microsoft, Samsung)

สำคัญที่ 3 ทำระบบที่เป็นการลงทุนเพื่ออนาคต เช่น Gemini AI, Quantum Computer (แบ่งกำไรส่วนหนึ่งมาลงทุนเพื่ออนาคต)

we provide a list of project characteristics that will affect the methodology selection decision.

- **Clarity of User Requirements** How well do the users and analysts understand the functions and capabilities needed from the new system?
- **Familiarity with Technology** How much experience does the project team have with the technology that will be used?
- **System Complexity** How much complexity is anticipated in the new system? Does the new system include a wide array of features? Will the system have to integrate with many existing systems? Does it span multiple organizational units, or even multiple organizations?
- **System Reliability** Will this system need to be highly reliable or is some downtime tolerable?
- **Short Time Schedules** Is the project time frame tight?
- **Schedule Visibility** Are the project sponsors, users, or organizational managers anxious to see progress?

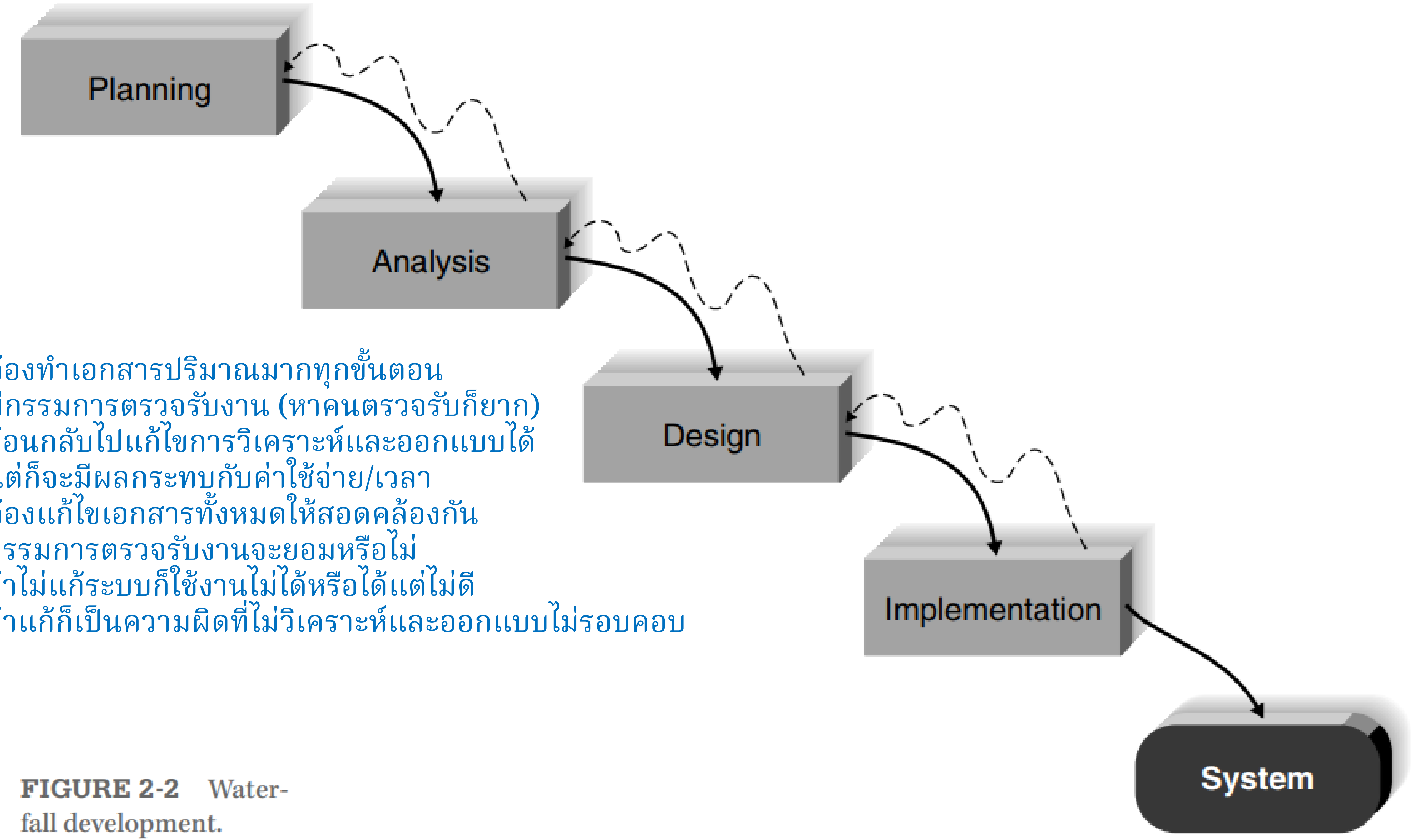
Waterfall Development

With **waterfall development** methodologies, the project team proceeds sequentially from one phase to the next (Figure 2-2). 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. Once the work produced in one phase is approved, the phase ends and the next phase begins. As the project progresses from phase to phase, it moves forward in the same manner as a waterfall. 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.)

การย้อนกลับไปแก้เฟสก่อนหน้าเล็ก ๆ น้อย ๆ ก็พอทำได้ แต่ก็เสียเวลาและค่าใช้จ่าย

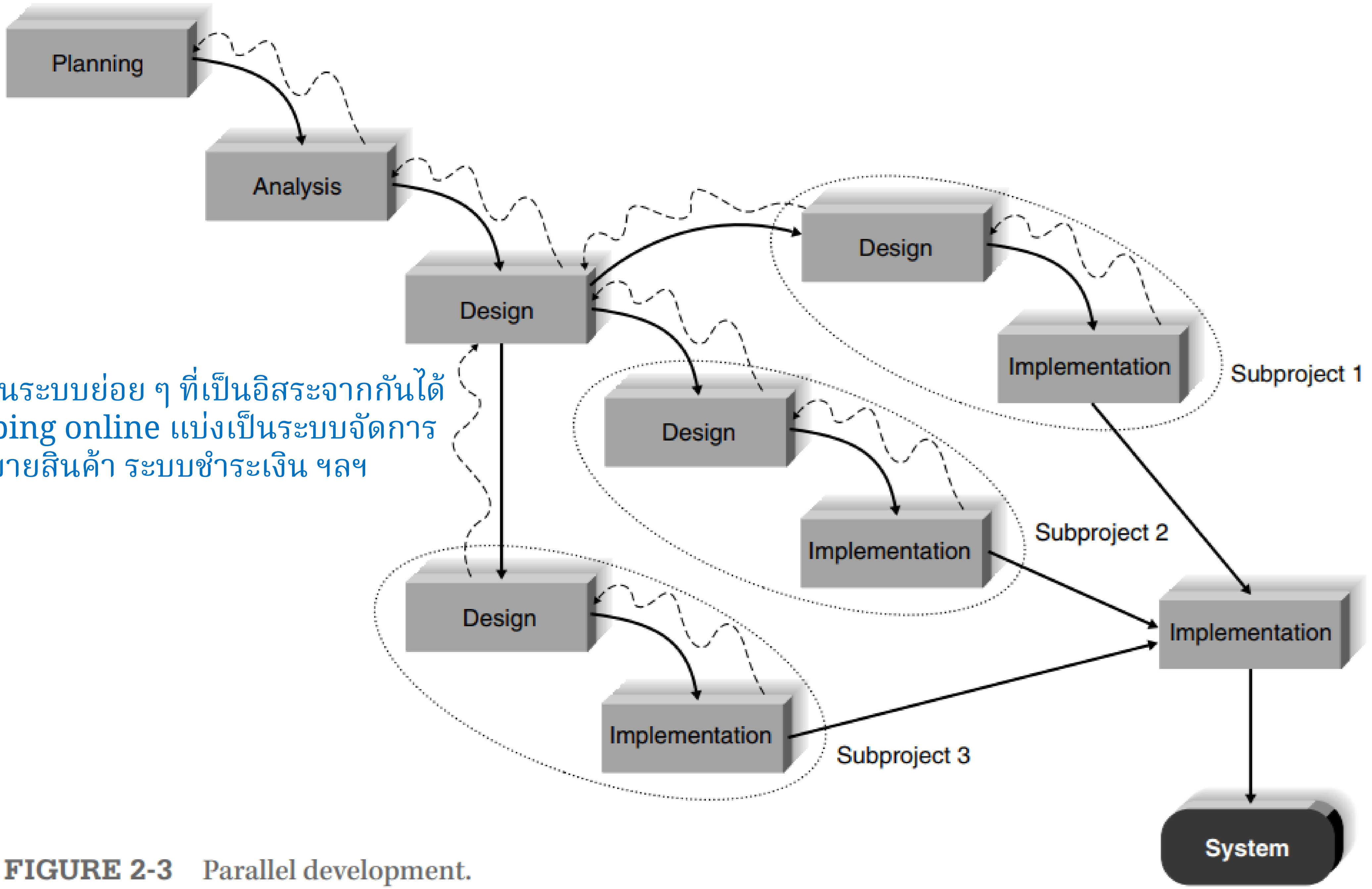
การแก้ไขเล็ก ๆ น้อย ๆ เช่น เพิ่มฟิลด์ข้อมูลเบอร์โทรศัพท์ของผู้ใช้

การแก้ไขที่มีผลกระทบมาก เช่น การเปลี่ยน requirements ที่ต้องแก้โครงสร้างฐานข้อมูล



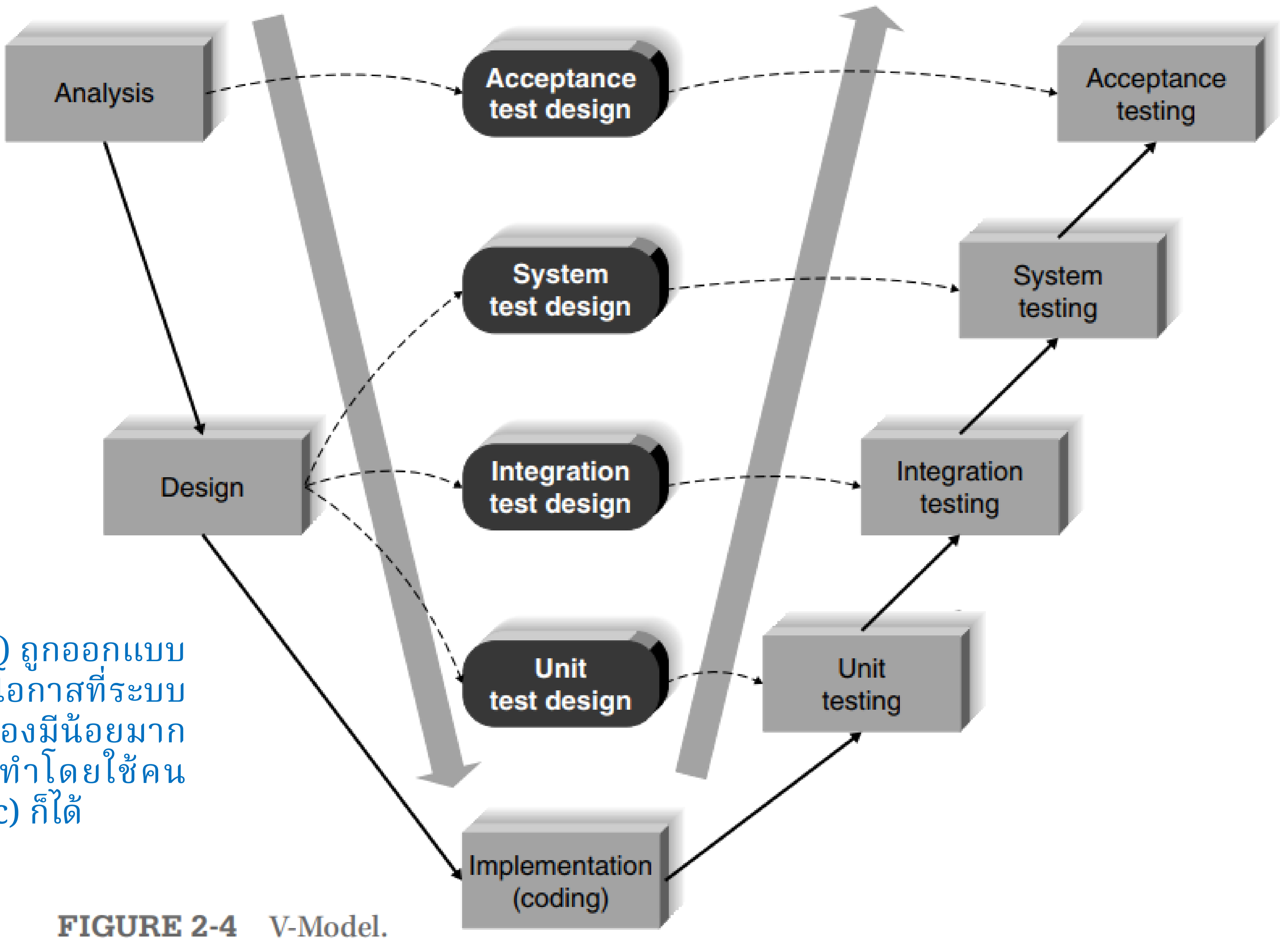
ต้องทำเอกสารปริมาณมากทุกขั้นตอน
มีกรรมการตรวจรับงาน (หาคนตรวจรับก็ยาก)
ย้อนกลับไปแก้ไขการวิเคราะห์และออกแบบได้
แต่ก็จะมีผลกระทบกับค่าใช้จ่าย/เวลา
ต้องแก้ไขเอกสารทั้งหมดให้สอดคล้องกัน
กรรมการตรวจรับงานจะยอมหรือไม่
ถ้าไม่แก้ระบบก็ใช้งานไม่ได้หรือได้แต่ไม่ดี
ถ้าแก้ก็เป็นความผิดที่ไม่วิเคราะห์และออกแบบไม่รอบคอบ

FIGURE 2-2 Waterfall development.



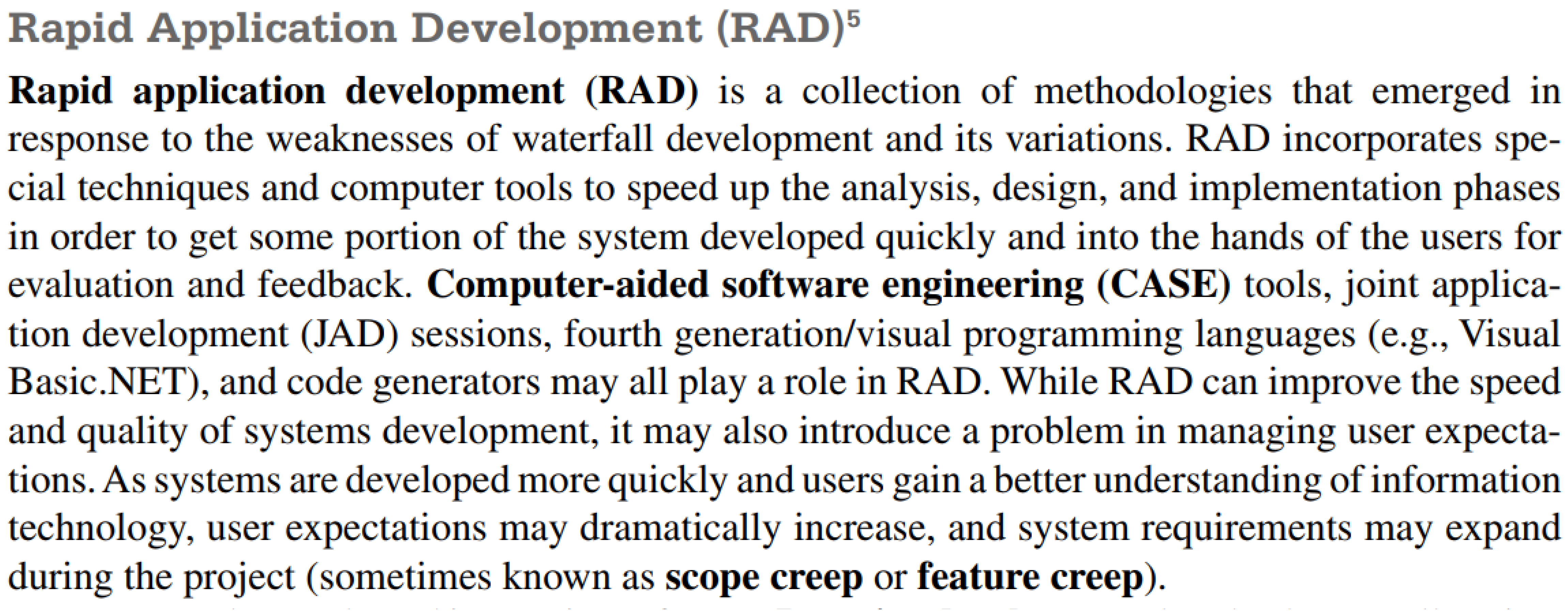
แบ่งงานเป็นระบบย่อย ๆ ที่เป็นอิสระจากกันได้
เช่น shopping online แบ่งเป็นระบบจัดการ
ผู้ใช้ ระบบขายสินค้า ระบบชำระเงิน ฯลฯ

FIGURE 2-3 Parallel development.



การทดสอบระบบ (testing) ถูกออกแบบโดยผู้เชี่ยวชาญตั้งแต่ต้น โอกาสที่ระบบจะทำงานผิดพลาด/บกพร่องมีน้อยมาก การทดสอบระบบอาจจะทำโดยใช้คน หรือโปรแกรม (automatic) ก็ได้

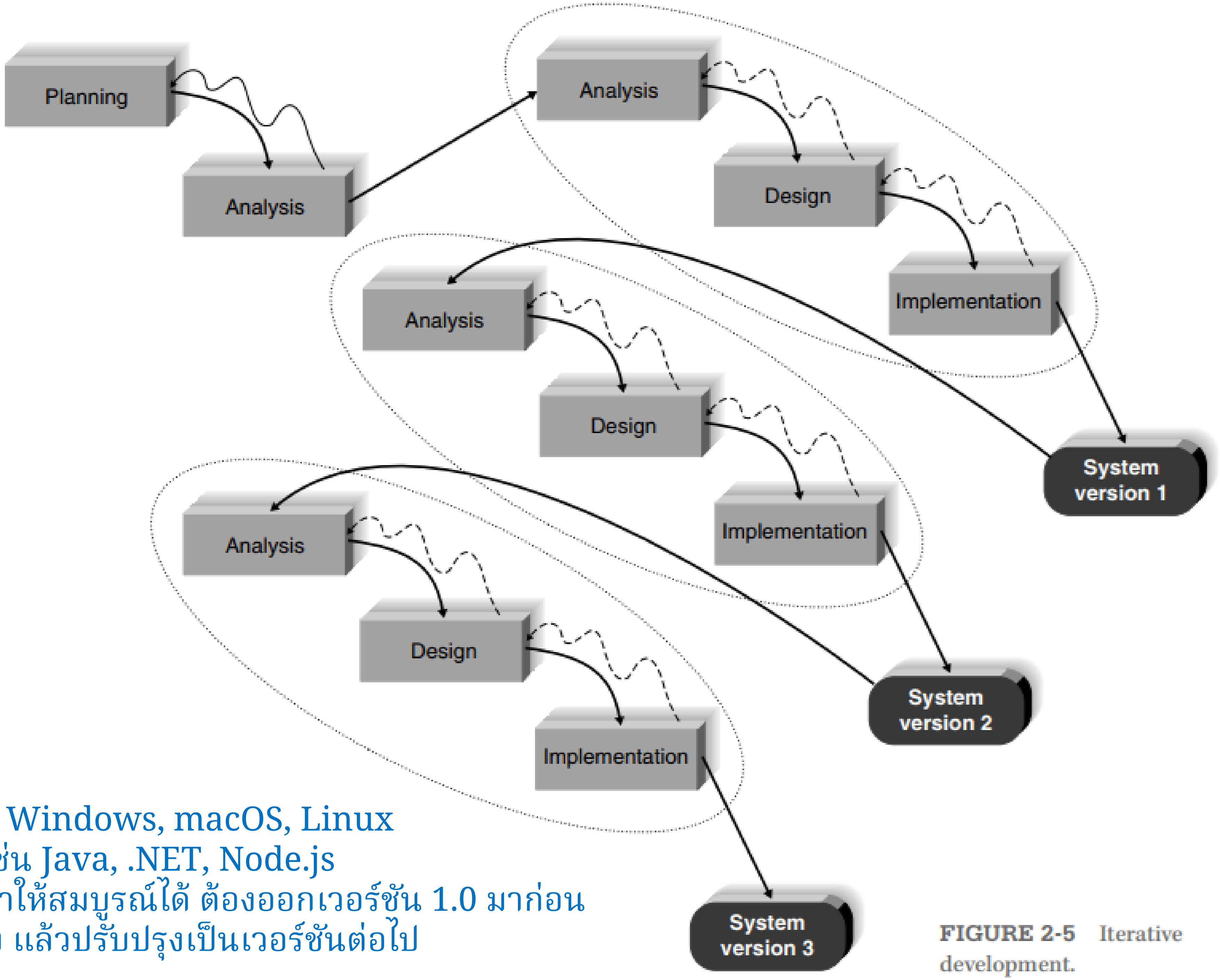
FIGURE 2-4 V-Model.



Rapid Application Development (RAD)[5]

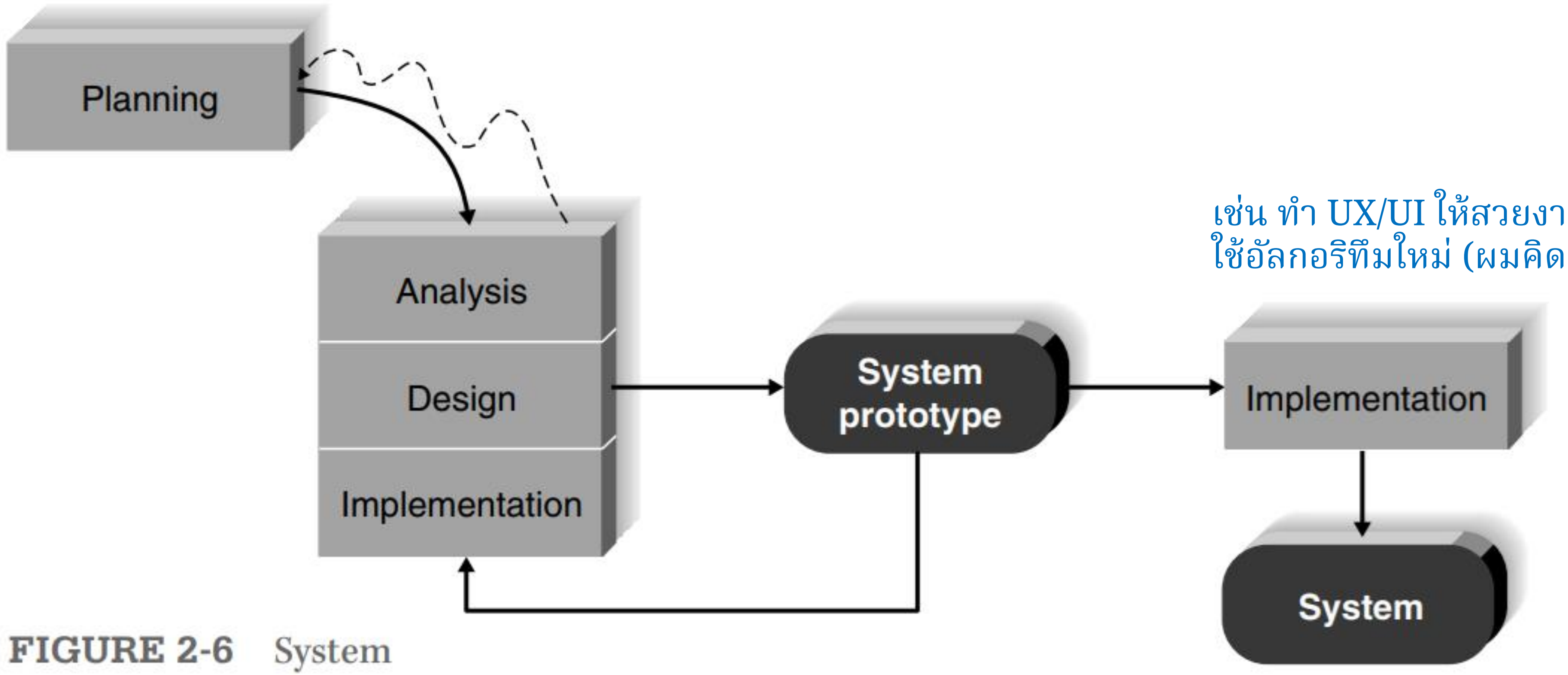
Rapid application development (RAD) is a collection of methodologies that emerged in response to the weaknesses of waterfall development and its variations. RAD incorporates special techniques and computer tools to speed up the analysis, design, and implementation phases in order to get some portion of the system developed quickly and into the hands of the users for evaluation and feedback. **Computer-aided software engineering (CASE)** tools, joint application development (JAD) sessions, fourth generation/visual programming languages (e.g., Visual Basic.NET), and code generators may all play a role in RAD. While RAD can improve the speed and quality of systems development, it may also introduce a problem in managing user expectations. As systems are developed more quickly and users gain a better understanding of information technology, user expectations may dramatically increase, and system requirements may expand during the project (sometimes known as **scope creep** or **feature creep**).

Scope Creep หมายถึง ขอบเขตของงานที่โดนขยายไปเรื่อย ๆ แบบไม่มีที่สิ้นสุด และเป็นสาเหตุหนึ่งที่ทำให้โครงการเกิดความล่าช้า และบางครั้งเป็นสาเหตุให้เกิดปัญหาข้อพิพาทกับผู้รับเหมา ผู้ว่าจ้างจะขยายขอบเขตงานได้แค่ไหน? ต้องประนีประนอมกับผู้รับเหมาด้วย ผู้รับเหมาอาจจะขอขยายเวลา/ขอค่าจ้างเพิ่ม ก็แล้วแต่จะตกลงกัน ผู้รับเหมาก็ไม่ควรประมูลงานมาในราคาต่ำเกินไป ต้องเผื่อการขยายขอบเขตไว้ด้วย



ตัวอย่างเช่น ระบบปฏิบัติการ Windows, macOS, Linux
เครื่องมือพัฒนาซอฟต์แวร์ เช่น Java, .NET, Node.js
เว็บเบราเซอร์ ฯลฯ ไม่มีทางทำให้สมบูรณ์ได้ ต้องออกเวอร์ชัน 1.0 มาก่อน
ทดสอบกับพฤติกรรมผู้ใช้จริง แล้วปรับปรุงเป็นเวอร์ชันต่อไป

FIGURE 2-5 Iterative development.



เช่น ทำ UX/UI ให้สวยงาม
ใช้อัลกอริทึมใหม่ (ผมคิดเองนะ)

FIGURE 2-6 System prototyping.

System prototyping performs the analysis, design, and implementation phases concurrently to quickly develop a simplified version of the proposed system and give it to the users for evaluation and feedback (Figure 2-6). The system prototype is a "quick and dirty" version of the system and provides minimal features.

Design prototype อาจจะทดลองทำ features ที่ผู้ใช้แนะนำมา ซึ่งผู้ใช้ก็ยังไม่ค่อยเข้าใจดีนัก หรือจินตนาการไม่ออกว่าถ้าทำไปแล้ว จะเป็นยังไง ดังนั้นทำเสร็จแล้วก็ทิ้งไปเลย เลือกเฉพาะบาง features ไปใช้

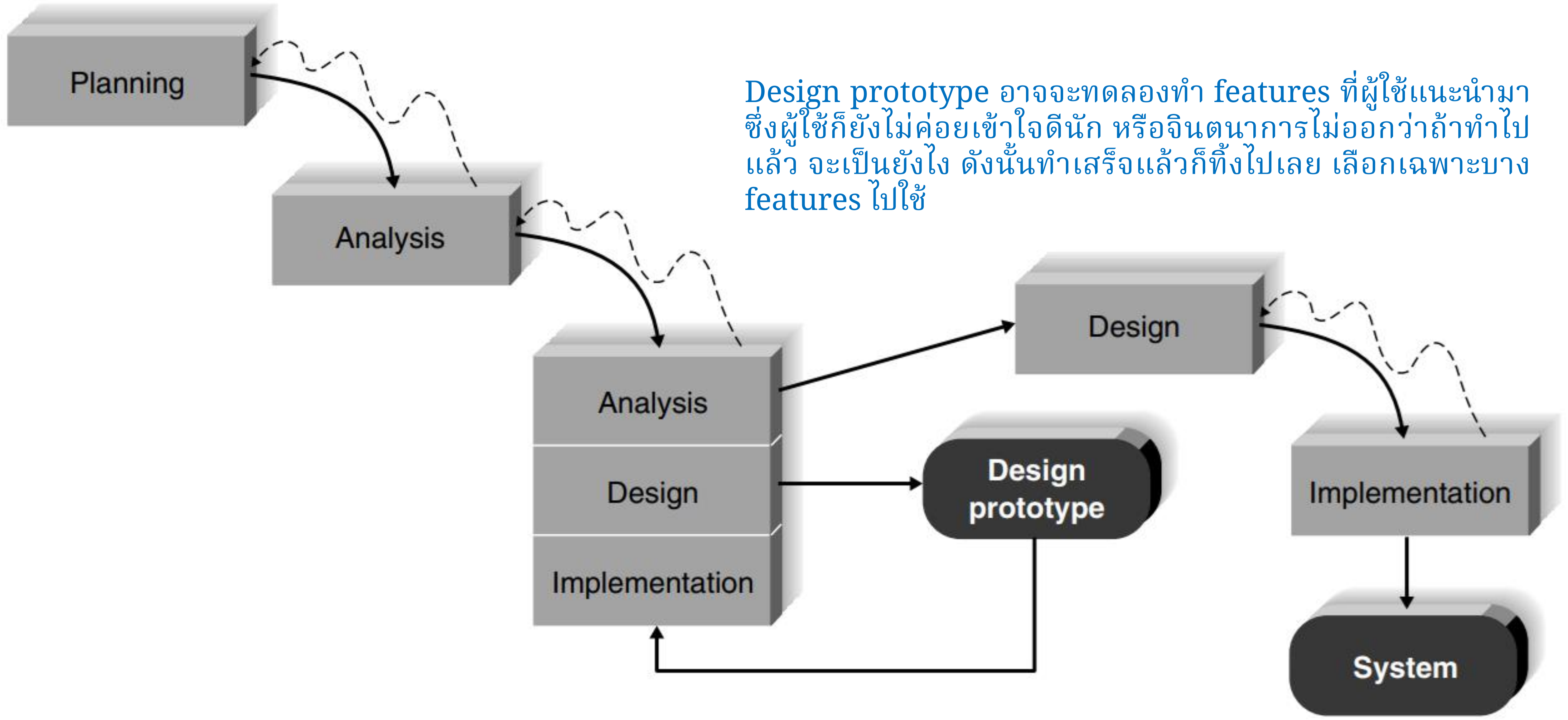


FIGURE 2-7 Throwaway prototyping.

Design prototype เช่น prototype car, fashion show ทดสอบแล้วก็ทิ้งไป ไม่ได้นำมาผลิตเพื่อขาย

Agile Development

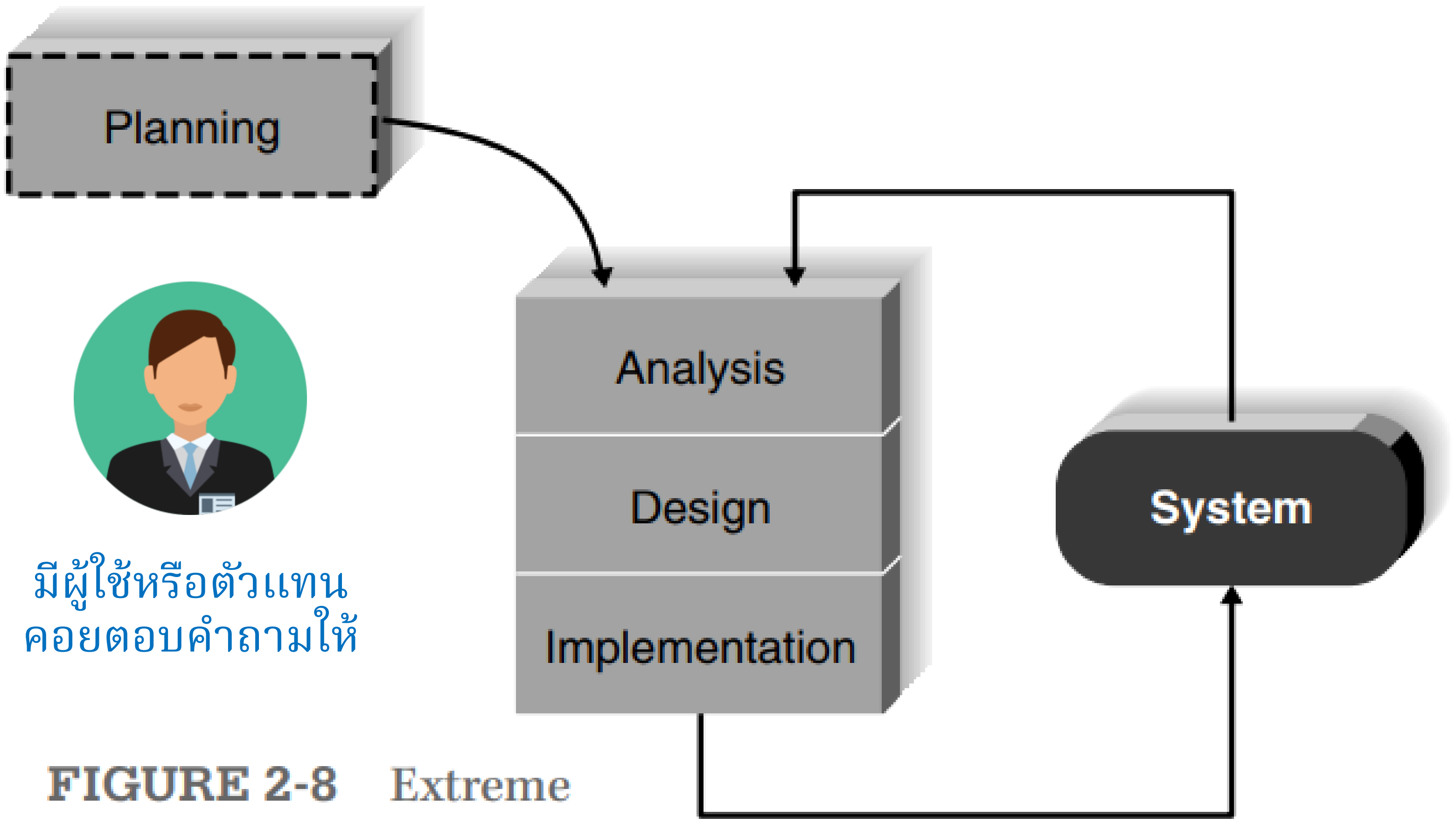
Agile development[7] is a group of programming-centric methodologies that focus on streamlining the SDLC. Much of the modeling and documentation overhead is eliminated; instead, face-to-face communication is preferred. A project emphasizes simple, iterative application development in which every iteration is a complete software project, including planning, requirements analysis, design, coding, testing, and documentation (Figure 2-8). Cycles are kept short (1–4 weeks), and the development team focuses on adapting to the current business environment. There are several popular approaches to agile development, including extreme programming (XP),[8] Scrum,[9] and dynamic systems development method (DSDM).[10] Here, we briefly describe XP. We expand our discussion of Agile development in Chapter 13.

XP[11] emphasizes customer satisfaction and teamwork. Communication, simplicity, feedback, and courage are core values. Developers communicate with customers and fellow programmers. Designs are kept simple and clean. Early and frequent testing provides feedback, and developers can courageously respond to changing requirements and technology. Project teams are kept small.

An XP project begins with user stories that describe what the system needs to do. Then, programmers code in small, simple modules and test to meet those needs. Users are required to be available to clear up questions and issues as they arise. Standards are particularly important to minimize confusion, so XP teams use a common set of names, descriptions, and coding practices. XP projects deliver results sooner than even the RAD approaches, and they rarely get bogged down in gathering requirements for the system.

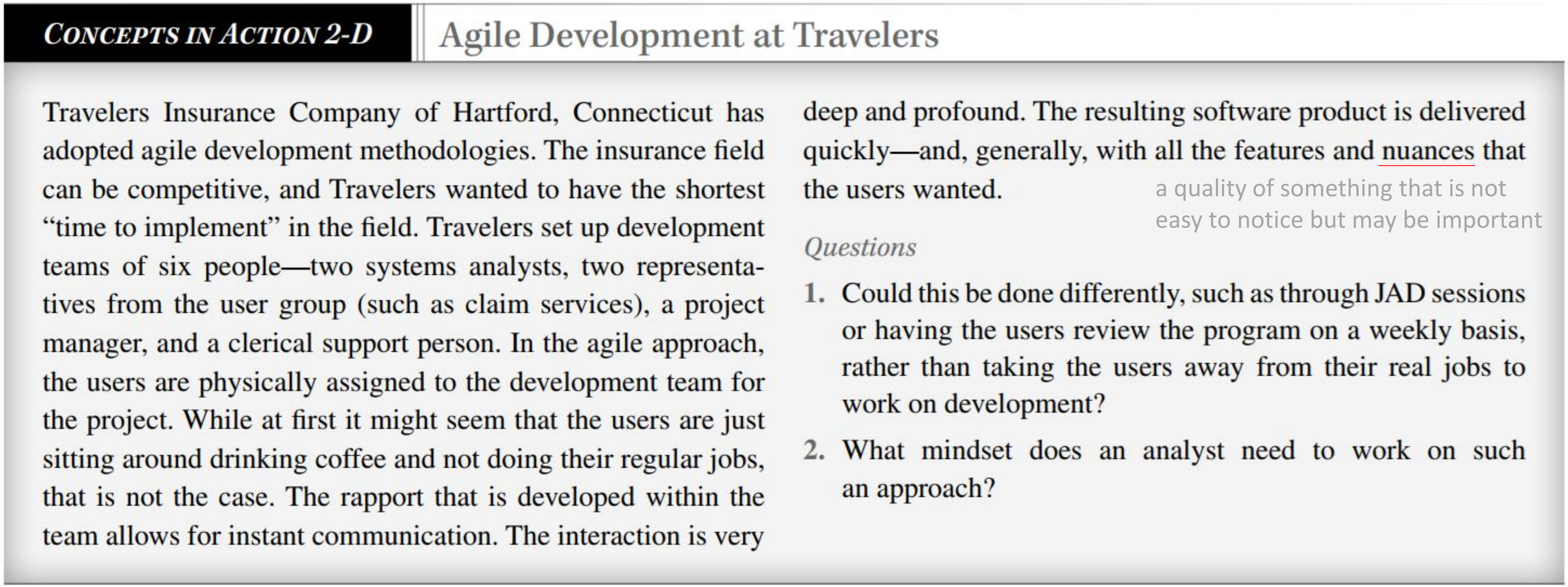
XP works well in projects with highly motivated, cohesive, stable, and experienced teams. If the project is not small or the teams are not jelled,[12] however, then the likelihood of success for the XP project is reduced. XP requires a great deal of discipline to prevent projects from becoming unfocused and chaotic. Furthermore, it is recommended only for small groups of developers (not more than 10), and it is not advised for mission-critical applications. Since little analysis and design documentation is produced with XP, there is only code documentation; therefore, maintenance of large systems developed using XP may be impossible. Also, since mission-critical business information systems tend to exist for a long time, the utility of XP as a business information system development methodology is in doubt. Finally, the methodology requires considerable on-site user involvement, something that is frequently difficult to obtain.[13]

[12] A "jelled team" is one that has low turnover, a strong sense of identity, a sense of eliteness, a feeling that they jointly own the product being developed, and enjoyment in working together. For more information regarding jelled teams, see T. DeMarco and T. Lister, *Peopleware: Productive Projects and Teams*, New York: Dorsett House, 1987.



มีผู้ใช้หรือตัวแทน
คอยตอบคำถามให้

FIGURE 2-8 Extreme programming.



CONCEPTS IN ACTION 2-D

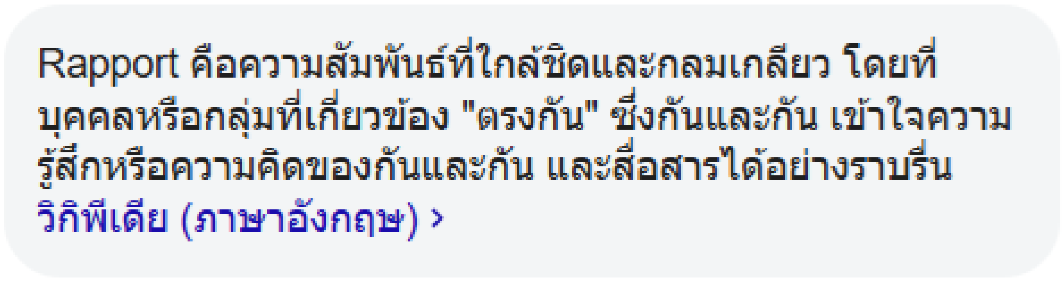
Agile Development at Travelers

Travelers Insurance Company of Hartford, Connecticut has adopted agile development methodologies. The insurance field can be competitive, and Travelers wanted to have the shortest "time to implement" in the field. Travelers set up development teams of six people—two systems analysts, two representatives from the user group (such as claim services), a project manager, and a clerical support person. In the agile approach, the users are physically assigned to the development team for the project. While at first it might seem that the users are just sitting around drinking coffee and not doing their regular jobs, that is not the case. The rapport that is developed within the team allows for instant communication. The interaction is very deep and profound. The resulting software product is delivered quickly—and, generally, with all the features and nuances that the users wanted.

a quality of something that is not easy to notice but may be important

Questions

1. Could this be done differently, such as through JAD sessions or having the users review the program on a weekly basis, rather than taking the users away from their real jobs to work on development?
2. What mindset does an analyst need to work on such an approach?



Rapport คือความสัมพันธ์ที่ใกล้ชิดและกลมเกลียว โดยที่บุคคลหรือกลุ่มที่เกี่ยวข้อง "ตรงกัน" ซึ่งกันและกัน เข้าใจความรู้สึกหรือความคิดของกันและกัน และสื่อสารได้อย่างราบรื่น
วิกิพีเดีย (ภาษาอังกฤษ) ›

Usefulness in Developing Systems	Waterfall	Parallel	V-Model	Iterative	System Prototyping	Throwaway Prototyping	Agile Development
With unclear user requirements	Poor	Poor	Poor	Good	Excellent	Excellent	Excellent
With unfamiliar technology	Poor	Poor	Poor	Good	Poor	Excellent	Poor
That are complex	Good	Good	Good	Good	Poor	Excellent	Poor
That are reliable	Good	Good	Excellent	Good	Poor	Excellent	Good
With short time schedule	Poor	Good	Poor	Excellent	Excellent	Good	Excellent
With schedule visibility	Poor	Poor	Poor	Excellent	Excellent	Good	Good

FIGURE 2-9 Criteria for selecting a methodology.

Clarity of User Requirements ผู้ใช้อาจจะพูดไม่เก่ง อธิบายไม่เป็น (non-IT)

When it is difficult to state the user requirements clearly, users may need to interact with technology to really understand what a new system can do and how to best apply it to their needs. System prototyping and throwaway prototyping are most appropriate when user requirements are unclear because they provide prototypes for users to interact with early in the SDLC. Agile development is suitable if on-site user involvement is available.

Familiarity with Technology

When the system will use new, unfamiliar technology, applying the new technology early in the methodology will improve the chance of success. Without some familiarity with the base technology, design risks increase. Throwaway prototyping is particularly appropriate for situations with limited familiarity with technology because it explicitly encourages the developers to create design prototypes for areas with high uncertainty. Iterative development is good as well, because opportunities are created to investigate the technology in some depth before the design is complete. While one might think that system prototyping would also be appropriate, it is much less so because the early prototypes that are built usually only scratch the surface of the new technology. Typically, it is only after several prototypes and several months that the developers discover weaknesses or problems in the new technology.

System Complexity

Complex systems require careful and detailed analysis and design. Throwaway prototyping is particularly well suited to such detailed analysis and design, but system prototyping is not. 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. 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.

System Reliability

System reliability is usually an important factor in system development. After all, who wants an unreliable system? For some applications, reliability is truly critical (e.g., medical equipment, missile control systems), while for other applications, it is merely important (e.g., games, Internet video). The V-model is useful when reliability is important, due to its emphasis on testing. Throw-away prototyping excels when system reliability is a high priority because detailed analysis and design phases are combined with the ability for the project team to test many different approaches through design prototypes before completing the design. System prototyping is generally not a good choice when reliability is critical, due to the lack of careful analysis and design phases that are essential to dependable systems.

Short Time Schedules

Projects that have short time schedules are well suited for RAD methodologies because those methodologies are designed to increase the speed of development. Iterative development, system prototyping, and agile methodologies are excellent choices when timelines are short because they best enable the project team to adjust the functionality in the system on the basis of a specific delivery date. If the project schedule starts to slip, it can be readjusted by removal of the functionality from the version or prototype under development. Waterfall-based methodologies are the worst choice when time is at a premium because they do not allow for easy schedule changes.

Concepts in Action 2-E Where Agile Works and Does Not Work

British Airways (BA) experienced problems in software development despite a willing and capable development team. Mike Croucher, brought in as chief software engineer, recommended a move to agile development after studying BA's development process. The movement to agile was carefully conducted, recognizing that agile represents a huge cultural shift for the developers. BA development team members who were amenable to and suitable for agile methods were trained as agile mentors and coaches to help ease the transition.

Converting to agile methods enabled BA to substantially shorten the time requirements of certain projects. In some cases, a project that might have taken 9 months following a traditional methodology was completed in 8 weeks. Only about 25% of the organization has changed to agile, however. BA recognized a continuing role for the waterfall methodology in certain areas of the organization and does not intend to force-fit agile everywhere. At BA, agile is used when the user base demands speed, flexibility, and customer-oriented design. Agile is ideal when an area requiring small functionality can be developed and deployed earlier, according to Croucher.

Adapted from: Mondelo, Daniel J. "Where agile development works and where it doesn't: A user story." SearchSoftwareQuality.com. February 24, 2010.

YOUR TURN 2-2 Selecting a Methodology

Suppose that you are an analyst for the ABC Company, a large consulting firm with offices around the world. The company wants to build a new knowledge management system that can identify and track the expertise of individual consultants anywhere in the world based on their education and the various consulting projects on which they have worked. Assume that this is a new idea that has never been attempted in ABC or elsewhere. ABC has an international network, but the offices in each country may use somewhat different hardware and software. ABC management wants the system up and running within a year.

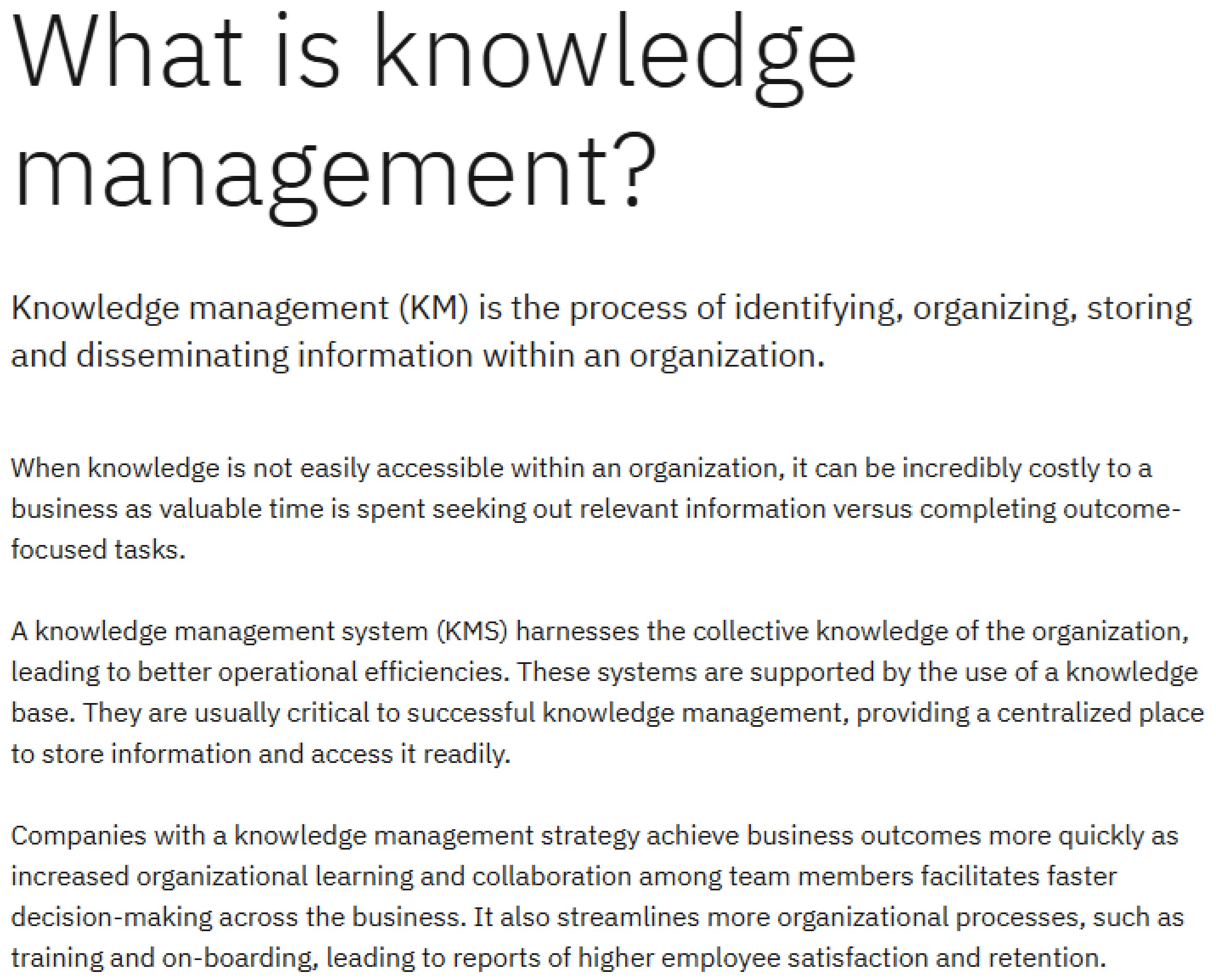
Question

1. What methodology would you recommend that ABC Company use? Why?

Usefulness in Developing Systems	Waterfall	Parallel	V-Model	Iterative	System Prototyping	Throwaway Prototyping	Agile Development
With unclear user requirements	Poor	Poor	Poor	Good	Excellent	Excellent	Excellent
With unfamiliar technology	Poor	Poor	Poor	Good	Poor	Excellent	Poor
That are complex	Good	Good	Good	Good	Poor	Excellent	Poor
That are reliable	Good	Good	Excellent	Good	Poor	Excellent	Good
With short time schedule	Poor	Good	Poor	Excellent	Excellent	Good	Excellent
With schedule visibility	Poor	Poor	Poor	Excellent	Excellent	Good	Good

FIGURE 2-9 Criteria for selecting a methodology.

ให้เลือก method จากในตารางนี้ เท่านั้น พร้อมทั้งให้เหตุผลประกอบ



What is knowledge management?

Knowledge management (KM) is the process of identifying, organizing, storing and disseminating information within an organization.

When knowledge is not easily accessible within an organization, it can be incredibly costly to a business as valuable time is spent seeking out relevant information versus completing outcome-focused tasks.

A knowledge management system (KMS) harnesses the collective knowledge of the organization, leading to better operational efficiencies. These systems are supported by the use of a knowledge base. They are usually critical to successful knowledge management, providing a centralized place to store information and access it readily.

Companies with a knowledge management strategy achieve business outcomes more quickly as increased organizational learning and collaboration among team members facilitates faster decision-making across the business. It also streamlines more organizational processes, such as training and on-boarding, leading to reports of higher employee satisfaction and retention.

https://www.ibm.com/think/topics/knowledge-management

Types of knowledge

The definition of knowledge management also includes three types of knowledge—tacit, implicit, and explicit knowledge. These types of knowledge are largely distinguished by the codification of the information.

- **Tacit knowledge:** This type of knowledge is typically acquired through experience, and it is intuitively understood. As a result, it is challenging to articulate and codify, making it difficult to transfer this information to other individuals. Examples of tacit knowledge can include language, facial recognition, or leadership skills.
- **Implicit knowledge:** While some literature equivocates implicit knowledge to tacit knowledge, some academics break out this type separately, expressing that the definition of tactic knowledge is more nuanced. While tacit knowledge is difficult to codify, implicit knowledge does not necessarily have this problem. Instead, implicit information has yet to be documented. It tends to exist within processes, and it can be referred to as "know-how" knowledge.
- **Explicit knowledge:** Explicit knowledge is captured within various document types such as manuals, reports, and guides, allowing organizations to easily share knowledge across teams. This type of knowledge is perhaps the most well-known and examples of it include knowledge assets such as databases, white papers, and case studies. This form of knowledge is important to retain intellectual capital within an organization as well as facilitate successful knowledge transfer to new employees.