

مهندسی نرم افزار

DR. TAEI ZADEH

Course

- ▶ Assessment
 - ▶ 20%
- ▶ Class activity
 - ▶ 10%
- ▶ Project and presentation
 - ▶ 20
- ▶ Exam
 - ▶ 50%

Software Engineering

A Practitioner's Approach

Seventh Edition

Roger S. Pressman

O'REILLY®

Software Engineering at Google

Lessons Learned
from Programming
Over Time



برنامه درس

	معرفی مهندسی نرم افزار (تاریخچه)	
	مهندسی نیازمندیهای نرم افزار	
	مهندسی نیازمندیهای نرم افزار	
	روشهای مرسوم مهندسی نرم افزار	
	ابزارهای مهندسی نرم افزار UML	
	ابزارهای مهندسی نرم افزار UML	
WBS-PMBOK - گانت چارت	مدیریت پروژه های نرم افزاری - مفاهیم	
	مدیریت پروژه های نرم افزاری - مفاهیم	
Click up - Trello	مدیریت پروژه های نرم افزاری ابزار	
	مهندسی نرم افزار ابری	
GIT	مدیریت ورژن در نرم افزار	

Chapter 1

► Software & Software Engineering

Slide Set to accompany
Software Engineering: A Practitioner's Approach, 7/e
by Roger S. Pressman
AND
Other Sources

نرم افزار چیست؟

Software is:

- (1) *دستورات instructions (computer programs) that when executed provide desired features, function, and performance;*
- (2) *ساختار داده data structures that enable the programs to adequately manipulate information and*
- (3) *مستندات documentation that describes the operation and use of the programs.*

مشخصه های نرم افزار

- ▶ ویژگی هایی وجود دارد که نرم افزار را از سایر محصولات انسانی متمایز می سازد. در ساخت سخت افزار تمامی مراحل ساخت (شامل تحلیل، طراحی، ساخت و تست) به فرم فیزیکی انجام می شود.
- ▶ نرم افزار فیزیکی نیست، بلکه قطعه ای منطقی است.
- ▶ گاه درون یک محصول سخت افزاری مانند تلفن همراه
- ▶ گاه درون یک محیط عملیاتی مانند بانک

تفاوت های ساخت نرم افزار و سخت افزار

- ▶ *1. Software Is Developed Or Engineered, It Is Not Manufactured In The Classical Sense.*

▶ نرم افزار توسعه می یابد

تفاوت های ساخت نرم افزار و سخت افزار

► 2. *Software Doesn't "Wear Out."*

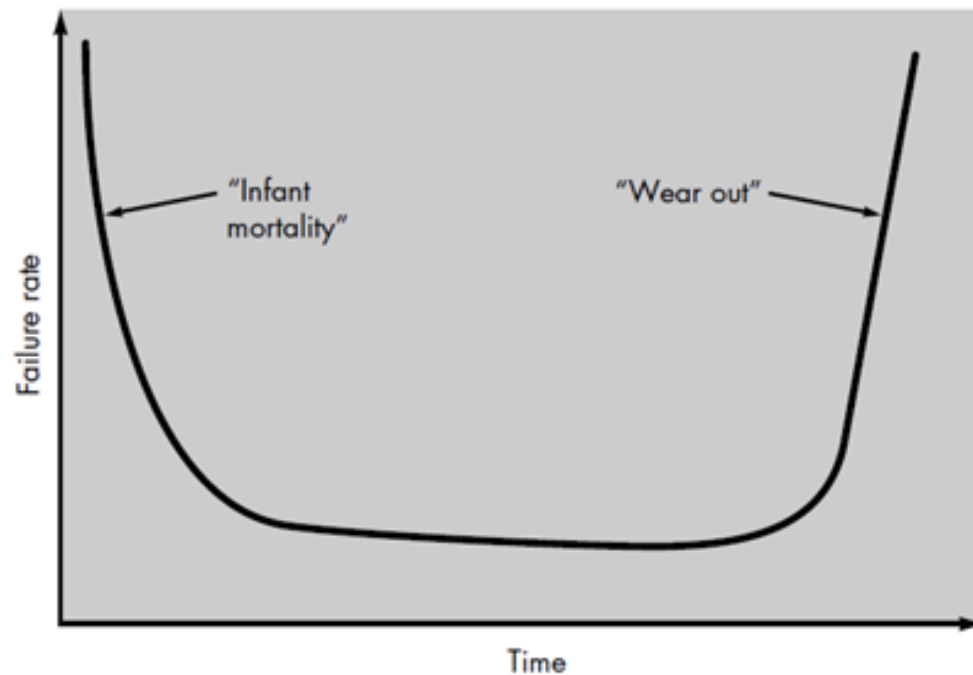
► سخت افزار در مراحل اولیه ساخت خطاهای زیادی دارد (به دلیل وجود خطا در مراحل طراحی و ساخت)، پس از اصلاح خطاها، در برهه ای از زمان، به کم ترین درجه خطا می رسد، ولی به تدریج شروع به فرسودگی می کند. زیرا به دلیل وجود خاک، دمای بالا و ... عمر مفید قطعات به پایان می رسد.

► نرم افزار فرسوده نمی شود

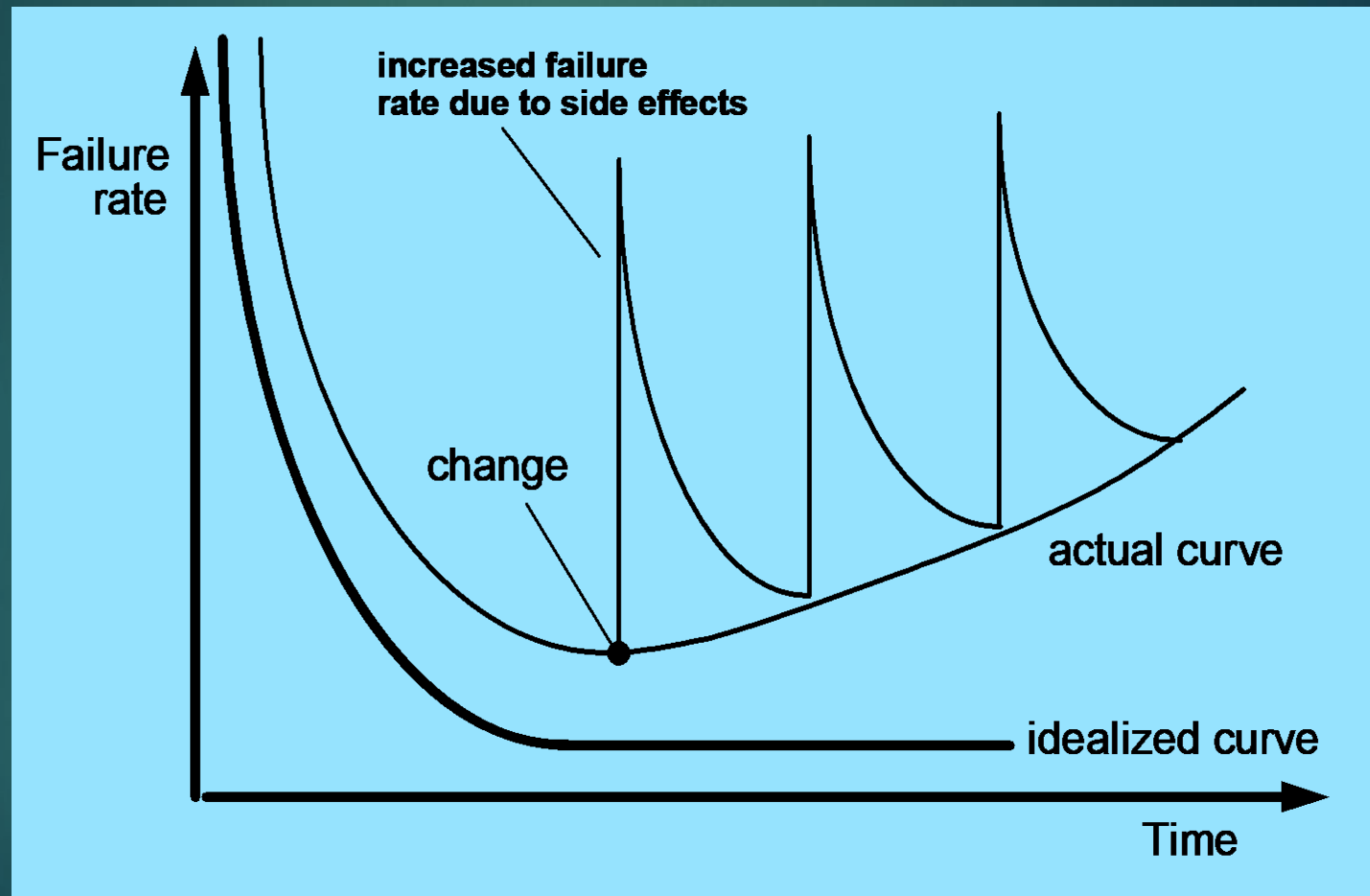
► ولی در نرم افزار مفهومی به نام Deterioration (زوال) داریم.

تفاوت های ساخت نرم افزار و سخت افزار

FIGURE 1.1
Failure curve
for hardware



منحنی ایده آل حیات نرم افزار



تفاوت های ساخت نرم افزار و سخت افزار

- ▶ *3. Although The Industry Is Moving Toward Component-based Construction, Most Software Continues To Be Custom-built.*
 - ▶ a software component should be designed and implemented so that it can be reused in many different programs.
 - ▶ today's graphical user interfaces are built using reusable components that enable the creation of **graphics windows, pull-down menus**, and a wide variety of interaction mechanisms. the data structure and processing detail required to build the interface are contained with a library of reusable components for interface construction.

مونتاژ قطعات:

در صنعت سخت افزار استفاده از کامپوننت های از پیش ساخته شده، امری معمول است و تمرکز روی طراحی وسایل جدید است ولی در صنعت نرم افزار، کم تر از مازول های آماده استفاده می شود.

بحران نرم افزار

دلایل

- ▶ هزینه های زیادی در تولید نرم افزار صرف شده است.
- ▶ نرم افزار تولید شده تمام نیازهای مشتریان را برآورده نکرده است.
- ▶ تحویل به موقع نرم افزار امکان پذیر نبود.
- ▶ توسعه سخت افزار بسیار سریع بوده و نرم افزار نتوانسته با آن رقابت کند.
- ▶ نرم افزار دارای خطاهای زیادی بوده و رفع آنها مشکل بوده است. قابلیت های توسعه نرم افزار دارای نگهداری و پشتیبانی بسیار محدودی بوده است.

نرم افزار موفق

▶ زمان

▶ بودجه پیش بینی شده

▶ نیازمندی واقعی کاربر

مهندسی نرم افزار

- ▶ تعریف
- ▶ ویژگی یکپارچگی در چرخه مهندسی نرم افزار
- ▶ فرایندها
- ▶ روش ها
- ▶ ابزارها
- ▶ بازه زمانی برنامه ریزی شده، بودجه پیش بینی شده و مطابق نیازهای واقعی
- ▶ ساخت یافته (فرآیند، روش، ابزارها) تابع
- ▶ شی گرا (فرآیند، روش، ابزار) کلاس

مهندس نرم افزار

- ▶ مسئول در برابر مشتری
- ▶ در پی دستیابی به ارزش مورد نیاز مشتری
- ▶ حل خواسته های مشتری
- ▶ فردی است که با استفاده از فرآیندها، ابزارها و روش های موجود به کمک علم و دانش خود و پس از تحلیل مساله آن را پیاده سازی و مدیریت می کند.
- ▶ مدیریت: (بازه زمانی از قبل پیش بینی شده، بودجه ای از قبل پیش بینی شده)

مهندس نرم افزار \neq برنامه نویس

برنامه نویسی جزئی از کارهای مهندس نرم افزار است.

کیفیت محصول = میزان رضایت مندی مشتری

مهندسی ساز یا غیر مهندسی ساز

▶ بحران نرم افزار

▶ شاخص مهندسی ساز:

▶ مستندات

▶ قابلیت توسعه

▶ رضایت مشتری

مستندات :

راهنما

ساخت: (توسعه و نگهداری) راهنمای پشتیبانی

مولفه های نرم افزاری

▶ ساختار داده ای

▶ عملکرد

▶ مستندات

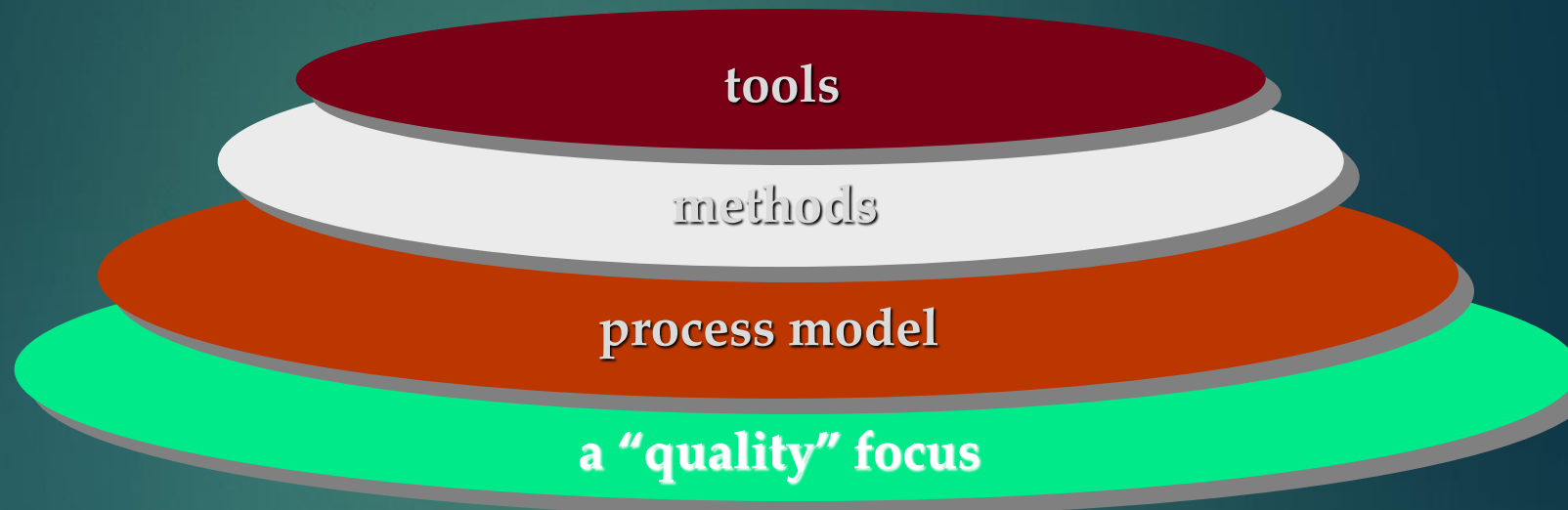
خروجی غیر مورد انتظار:
علت : ابزار، روش یا فرایند؟؟

انواع نرم افزار

- ▶ کاربردی
- ▶ سیستمی
- ▶ بی درنگ
- ▶ مدیریت پایگاه داده
- ▶ هوش مصنوعی
- ▶ متن باز

- ▶ فعالیت‌های چارچوب فرآیند مهندسی نرم‌افزار توسط تعدادی فعالیت چتری تکمیل می‌شوند. عبارتند از:
- **پیگیری و کنترل پروژه نرم‌افزار:** به تیم نرم‌افزار اجازه می‌دهد تا پیشرفت را در برابر برنامه پروژه ارزیابی کرده و هر اقدام لازم برای حفظ برنامه را انجام دهد.
- **مدیریت ریسک:** خطراتی را که ممکن است بر نتیجه پروژه یا کیفیت محصول تأثیر بگذارند، ارزیابی می‌کند.
- **تضمین کیفیت نرم‌افزار:** فعالیت‌های مورد نیاز برای اطمینان از کیفیت نرم‌افزار را تعریف و انجام می‌دهد.
- **بررسی‌های فنی:** محصول مهندسی نرم‌افزار را برای کشف و حذف خطاها قبل از انتشار آنها به فعالیت بعدی ارزیابی می‌کند.
- **اندازه‌گیری:** معیارهای فرآیند، پروژه و محصول را تعریف و جمع‌آوری می‌کند که به تیم کمک می‌کند نرم‌افزاری ارائه دهد که نیازهای ذینفعان را برآورده کند؛ می‌تواند در ارتباط با تمام فعالیت‌های چارچوب و چتری دیگر استفاده شود.
- **مدیریت پیکربندی نرم‌افزار:** اثرات تغییر در طول فرآیند نرم‌افزار را مدیریت می‌کند.
- **مدیریت قابلیت استفاده مجدد:** معیارهایی را برای استفاده مجدد از محصول (از جمله اجزای نرم‌افزار) تعریف کرده و مکانیزم‌هایی را برای دستیابی به قابلیت استفاده مجدد ایجاد می‌کند.
- **آماده‌سازی و تولید:** فعالیت‌های مورد نیاز برای ایجاد محصول مانند مدل‌ها، اسناد، گزارش‌ها، فرم‌ها و لیست‌ها را در بر می‌گیرد.

A Layered Technology



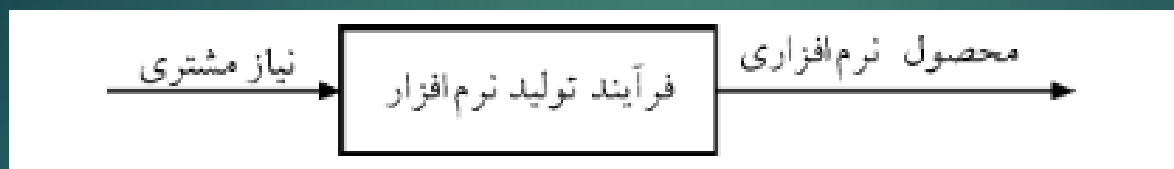
Software Engineering



► برآورده ساختن نیاز مورد انتظار مشتری

► شیوه رسیدن به انتظار مشتری: ابزار و روش های متفاوت دارد

► الگو و قالب تعریف کننده نحوه طی مراحل مختلف یک پروژه



► چه کسی، چه کاری را چه زمانی انجام دهد. هماهنگی در تعریف بین افراد تیم

► انتخاب فرآیند بستگی به ماهیت نرم افزار دارد.

روش ها

ساخت یافته

چه عملکردهایی داریم و این عملکردها به چه داده هایی نیاز دارند

شی گرا

چه داده هایی داریم و این داده ها چه عملکردهایی دارند (داده و عملکرد در قالب یک کلاس)

قابلیت استفاده مجدد reusability

برش و تکه تکه کردن نرم افزار decomposition

زمان و هزینه از پیش تعیین شده

ابزارها

- ▶ ساخت یافته
- ▶ E-r, **bpmn**, **uml**
- ▶ شی گرا UML
- ▶ Unified modeling language
- ▶ Case tool: در مراحل طراحی، پیاده سازی و تست و استقرار

- ▶ نحوه ارتباط معماری چهار لایه با هم را مشخص میکند.
- ▶ ساخت یافته
- ▶ مدل سازی داده و عملکرد
- ▶ متدلوژی SSADM مرسوم ترین و مدل فرآیند تولید آبشاری و ابزارهای ساخت یافته می باشد.
- ▶ شی گرا
- ▶ مدلسازی بر اساس کلاس بصورت سلسله مراتبی وراثتی
- ▶ متدلوژی **RUP** مرسوم ترین

متدولوژی شی گرا

- ▶ مدل شی گرا
- ▶ روش شی گرا
- ▶ ابزارهای شی گرا
- ▶ در اولین مرحله مدلسازی سیستم در قالب کلاسها (صفت و متد) نشان داده می شود.

فرآیند تولید نرم افزار به روش ساخت یافته

Requirements engineering مهندسی نیازمندیهای مشتری ▶
ارتباطات ▶

تهیه لیست نیازمندی های مشتری ▶

درک : گپ و گفت اولیه ▶

استخراج: بیشتر و تفصیلی ▶

QFD

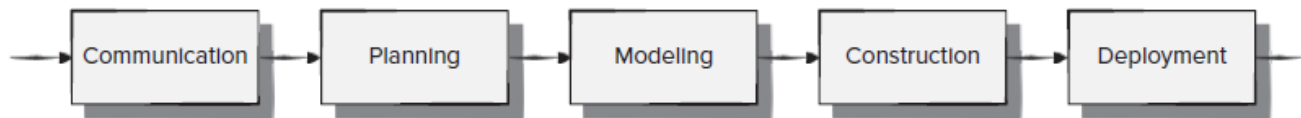
خواسته های عادی توسط مشتری... مورد انتظار مشتری بیان نمیکند... هیجان انگیز مشتری بیان نمیکند ▶

تشریح نیازمندی های شناخته شده - use case diagram ▶

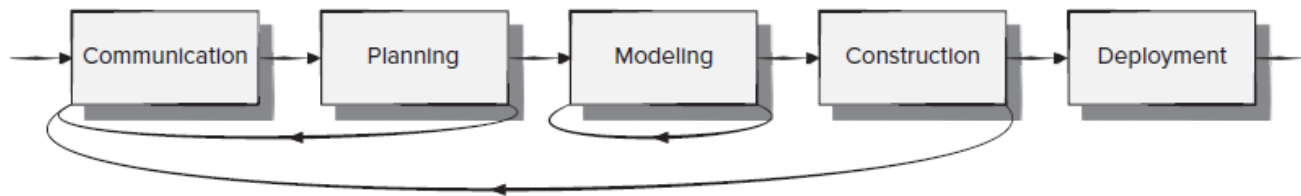
مذاکره ▶

اعتبار سنجی نیازمندیها : قابل پیاده سازی - ناسازگار - ▶

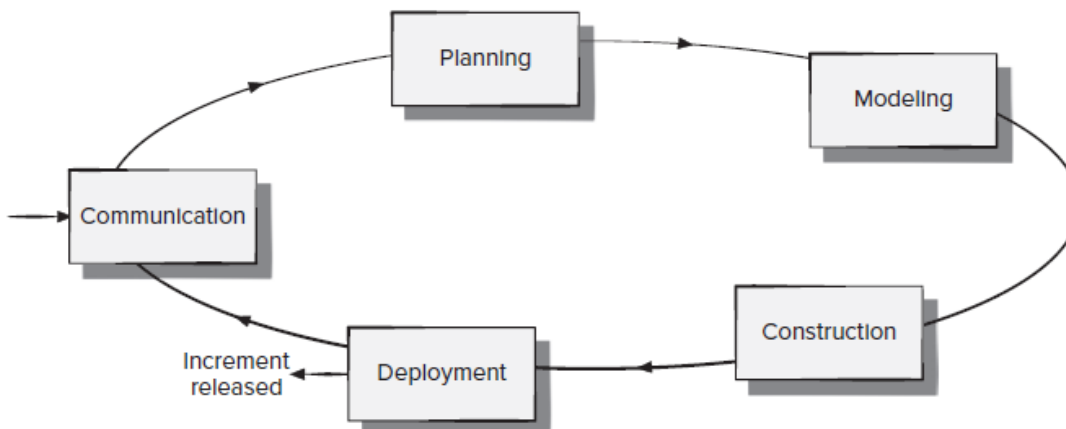
مدیریت نیازمندیها: نیازها همیشه در حال تغییر هستند ▶



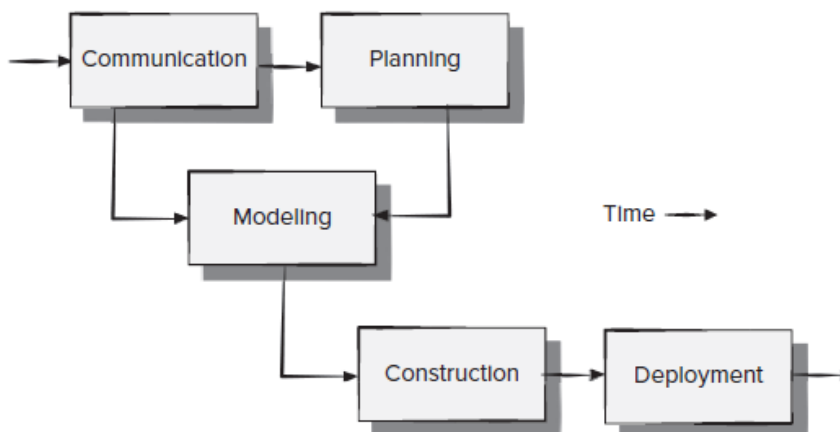
(a) Linear process flow



(b) Iterative process flow



(c) Evolutionary process flow



(d) Parallel process flow