2

ویژگی‌های کیفیتی

درس مهندسی نرم‌افزار

مفاهیم کلیدی

- انواع نیازمندی‌های نرم افزار
- خصوصیات کیفیتی
- معماری و خصوصیات کیفیتی
- طبقه‌بندی خصوصیات کیفیتی
- سناریوهای خصوصیات کیفیتی

نیازمندی‌های نرم‌افزار

- نیازمندی‌های وظیفه‌مندی (*Functional Requirements*)
  - وظایفی را که سیستم باید انجام دهد را توصیف می‌نمایند
- نیازمندی‌های غیروظیفه‌مندی (*Non-Functional Requirements*)
  - وظایفی که نحوهٔ انجام عملیات توسط سیستم را توصیف می‌نمایند
  - خصوصیات کیفیتی نرم‌افزار (*Quality Attributes*)

توسعه براساس نیازمندی‌ها

- چه وقت نیازمندی‌های غیروظیفه‌مندی به نرم‌افزار افزوده می‌شوند؟

تاثیر این دو نوع نیازمندي بر یکدیگر مشخص مي‌کند که چه وقت نیازمندي‌هاي غیروظیفه‌مندي باید افزوده شود

- یک روش قدیمی

ابتدا نرم‌افزاری می‌سازیم که خصوصیات وظیفه‌مندی را برآورده سازد و سپس خصوصیات غیر وظیفه‌مندی را به آن اضافه یا کم می‌کنیم

  - از دست دادن منابع و زمان
  - کیفیت پائین نرم‌افزار بخاطر عدم تاثیر این دو نیازمندی بریکدیگر

وظیفه‌مندی و معماری

- وظیفه‌مندی و خصوصیات کیفیتی متعامد هستند (در تئوری)
  - اما هر سطح از خصوصیات کیفیتی قابل انجام توسط هر سطح از وظیفه‌مندی نیست
- وظیفه‌مندی می‌تواند به روش‌های مختلف انجام شود، اما همه این روش‌ها معمارانه نیستند
- معماری ابزاری برای انجام خصوصیات کیفیتی
  - این کار با ساختاردهی به وظیفه‌مندی در قالب مولفه‌ها انجام می‌شود
- معمار سطح مناسبی از کیفیت را برای هر خصوصیت برمی‌گزیند

معماری و خصوصیات کیفیتی

- انجام خصوصیات کیفیتی در سراسر طراحی، پیاده‌سازی و استقرار می‌بایست در نظر گرفته شود
- خصوصیات کیفیتی هم دارای جنبه‌های معمارانه و هم غیرمعمارنه هستند. برای نمونه
  - قابلیت استفاده: انتخاب عناصر فرم در مقابل حمایت از undo (معمارانه)
  - قابلیت اصلاح‌پذیری: چگونگی تقسیم وظیفه‌مندی (معمارانه) و تکنیک‌های کدنویسی در یک ماژول
  - کارایی: میزان ارتباطات میان مولفه‌ها در مقابل کدنویسی الگوریتم‌ها

معماری و خصوصیات کیفیتی (ادامه)

- خصوصیات کیفیتی مستقل نیستند و نمی‌توانند به تنهایی انجام شوند
- همبستگی مثبت
  - قابلیت اصلاح‌پذیری در مقابل قابلیت ساخت (در بیشتر موارد)
- همبستگی منفی
  - قابلیت اعتماد در مقابل امنیت
  - کارایی در مقابل بقیه خصوصیات کیفیتی

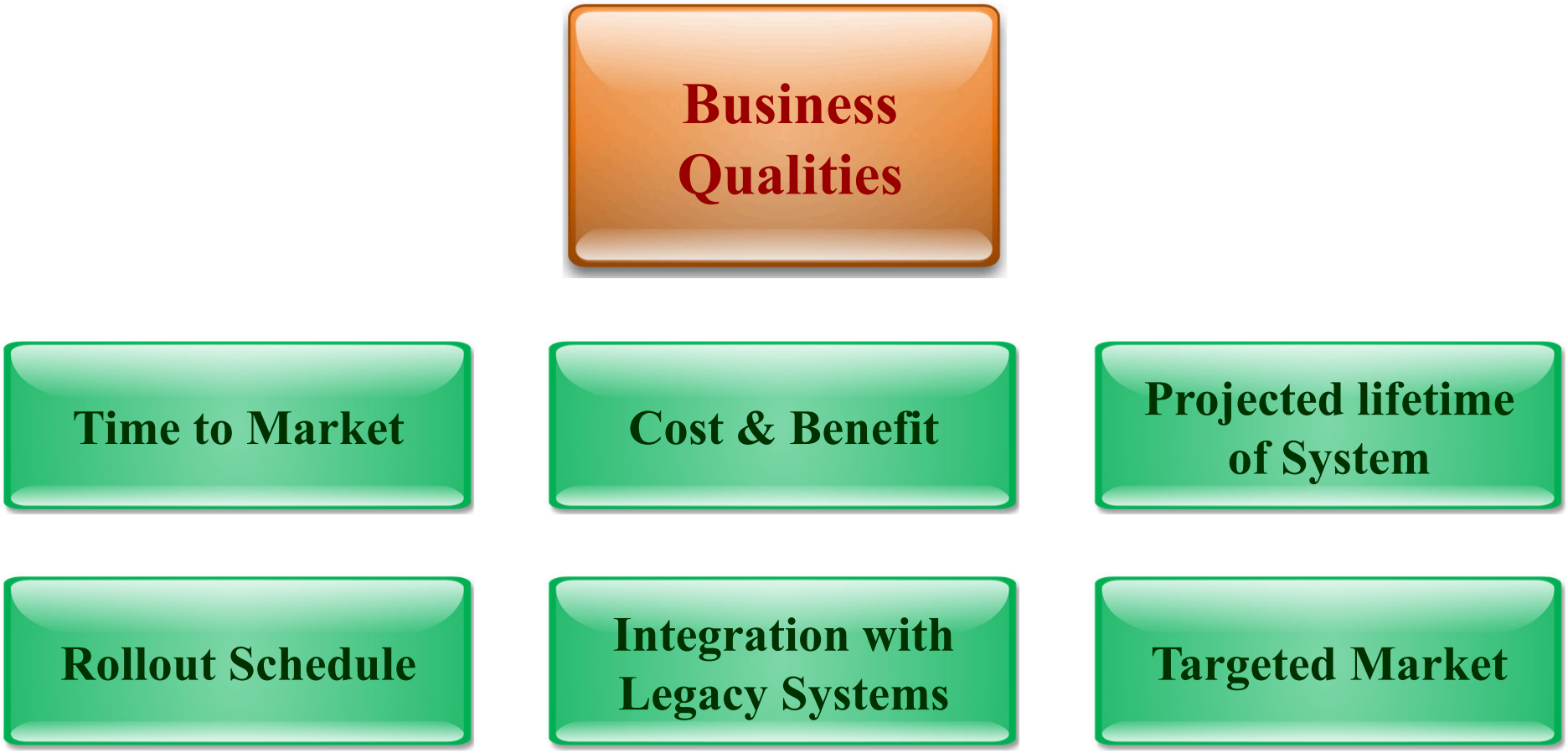
طبقه‌بندی خصوصیات کیفیتی

- خصوصیات کیفیتی سیستم
  - قابلیت دسترسی، قابلیت اصلاح، کارایی، امنیت، قابلیت تست، قابلیت استفاده و ...
- خصوصیات کیفیتی حرفه (معماری بر آنها تاثیر می‌گذارد)
  - زمان ارائه به بازار
- خصوصیات کیفیتی که معماری را تحت تاثیر قرار می‌دهند
  - جامعیت مفهومی

طبقه‌بندی خصوصیات کیفیتی (ادامه)

طبقه‌بندی خصوصیات کیفیتی (ادامه)

طبقه‌بندی خصوصیات کیفیتی (ادامه)

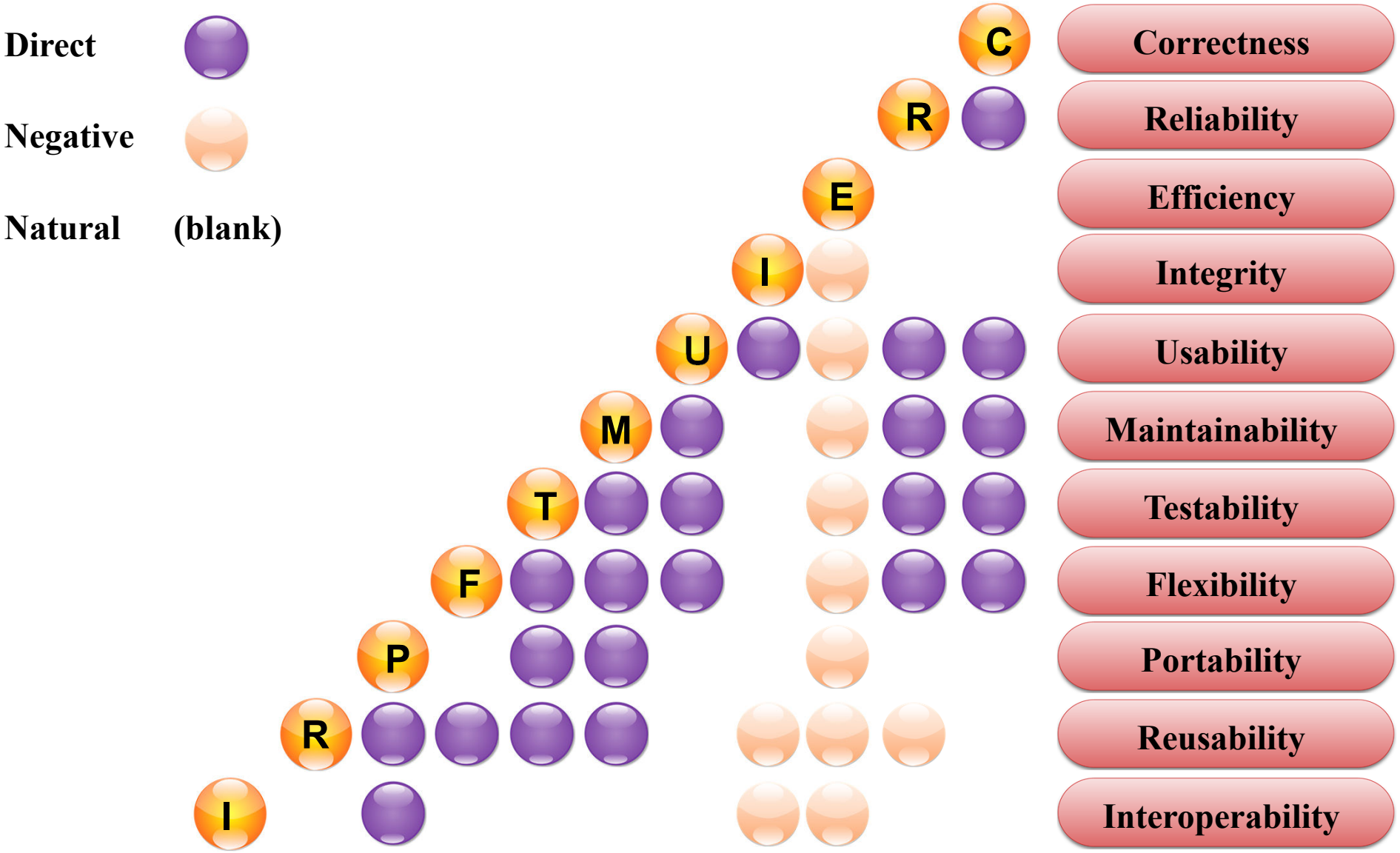


طبقه‌بندی خصوصیات کیفیتی (ادامه)

طبقه‌بندی خصوصیات کیفیتی (ادامه)

- قابل مشاهده از طریق اجرا
  - همانند کارایی و امنیت
- غیر قابل مشاهده از طریق اجرا
  - همانند قابلیت اصلاح‌پذیری و قابلیت تست
  - خصوصیات غیرقابل مشاهده برخی اوقات بسیار مهم می‌شوند
- طبقه‌بندی‌ها در مجموع مستقل (متعامد) هستند، اگر چه اعضای طبقه‌بندی دوم به صورت غیرمستقیم اعضای اولی را تحت تـاثیر قرار می‌دهند

همبستگی خصوصیات کیفیتی (McCall)



گذشته خصوصیات کیفیتی

- خصوصیات کیفیتی از دهه ۱۹۷۰ مورد توجه جامعه نرم‌افزاری قرار گرفته است
- نگاهی کوتاه به مشکلات گذشته
  - تعاریف یک خصوصیت عملیاتی نیست
    - قابلیت اصلاح‌پذیری با توجه به کدام جنبه؟
  - یک جنبه خاص به کدام خصوصیت کیفیتی تعلق دارد
    - آیا خطای سیستم یکی از جنبه‌های کارایی، امنیت یا قابلیت استفاده است؟
  - هر جامعه نرم‌افزاری فرهنگ اصطلاحات خود را توسعه داده است
    - جامعه کارایی «رویدادها»، جامعه امنیت «حملات»، جامعه قابلیت دسترسی «خطاها»– شاید همه به یک اتفاق اشاره کنند

سناریوهای خصوصیات کیفیتی

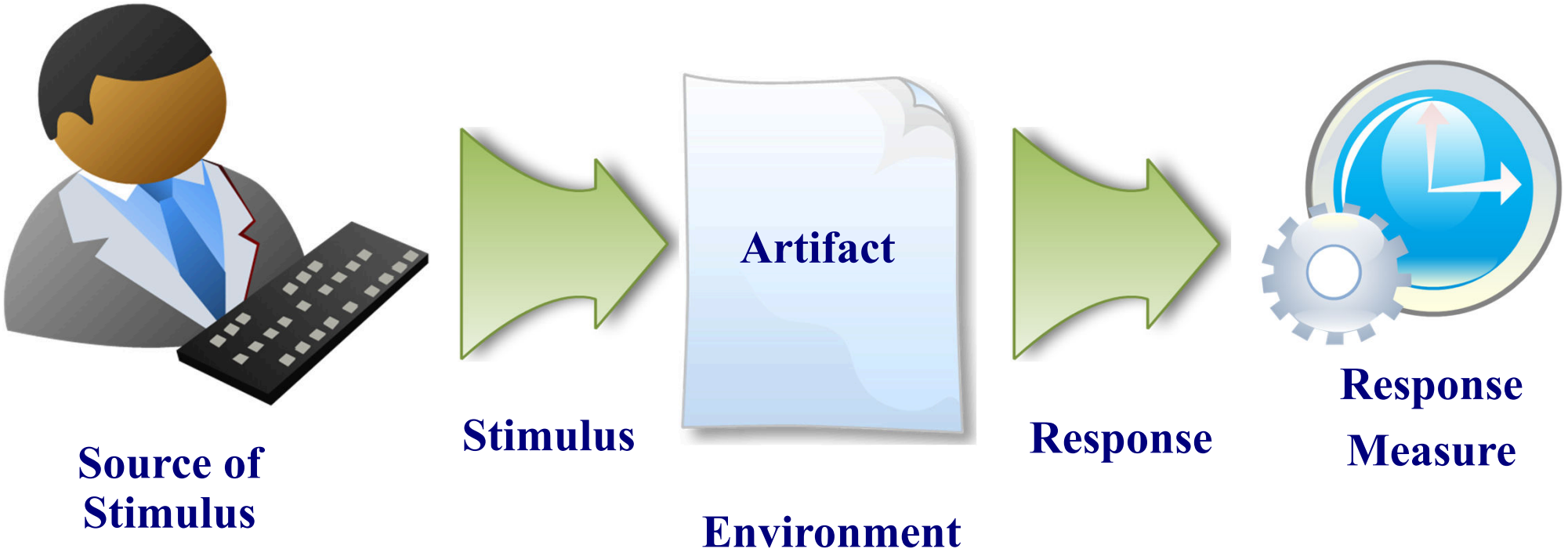
- آیا راه‌حلی برای مشکل بیان شده وجود دارد؟

- برای مشکلات اول و دوم می‌توان از سناریوهای خصوصیت کیفیتی استفاده نمود
- برای سومی می‌توان توصیف کوتاهی برای مفهوم هر خصوصیت کیفیتی ارائه نمود
- سناریوها می‌توانند عمومی یا منحصربفرد (برای سیستم خاص) باشند

سناریوهای خصوصیات کیفیتی (ادامه)

- نیازمندی مخصوص خصوصیت کیفیتی است که شامل
  - منبع تحریک: موجودیتی که تحریک را ایجاد می‌کند
  - محرک: شرطی که باید مورد توجه قرار گیرد
  - محیط: شرایطی که تحت آن محرک اتفاق می‌افتد (مثلاً سیستم با حجم بار بسیاری مواجه شده است)
  - فرآورده: قسمت‌هایی از سیستم که تحریک می‌شوند
  - پاسخ: واکنش مورد نظر بعد از رسیدن محرک
  - معیار پاسخ: پاسخ باید با استفاده از روش‌هایی قابل اندازه‌گیری باشد، تا نیازمندی بتواند آزمایش شود

بخش‌های خصوصیات کیفیتی



نمونه: سناریوی عمومی قابلیت دسترسی

Source:
Internal,
External

Stimulus:
(Fault)
Omission,
Crash,
Timing,
Response

Artifact
Process,
Storage,
Processor,
Communication

Environment:
Normal,
Degraded Operation

Response:
Record,
Notify,
Disable,
Continue,
(Normal/
Degraded) Be
Unavailable

Response Measure:
Repair Time,
Availability,
Available/
Degraded
Time Interval

سناریوهای خصوصیات کیفیتی در عمل

- سناریوهای عمومی چارچوبی را برای تولید تعداد زیادی سناریوی جامع، مستقل از سیستم و مخصوص به خصوصیات کیفیتی فراهم می‌کنند
- برای اینکه سناریوی عمومی برای یک سیستم خاص مفید باشد، باید آنها را *system-specific* نمائید
- *system-specific* نمودن یک سناریوی عمومی به معنی ترجمه آن به اصطلاحات عینی برای سیستم خاص است

سناریوهای خصوصیات کیفیتی در عمل (ادامه)

- هر سناریوی عمومی می‌تواند دارای چندین نسخه سناریوی *system-specific* داشته باشد
  - بعنوان نمونه، سیستمی که مجبور است از *Browser* جدیدی حمایت نماید، ممکن است مجبور باشد از رسانه جدید دیگری نیز حمایت نماید

قابلیت دسترسی

- قابلیت دسترسی با خطاهای سیستم و نتایج آن مرتبط است
- خطای سیستم وقتی اتفاق می‌افتد که سیستم دیگر خدمات بیان شده با خصوصیاتش را ارائه نمی‌دهد
  - برخی خطاها توسط کاربر (انسان یا سیستم) قابل مشاهده است
- تفاوت خطا و نقص
  - نقص می‌تواند به خطا تبدیل شود اگر اصلاح یا پوشش داده نشود
  - خطا بوسیله کاربران سیستم قابل مشاهده است
  - وقتی نقص قابل مشاهده می‌گردد که به خطا تبدیل می‌شود

قابلیت دسترسی (ادامه)

- قابلیت دسترسی یک سیستم
  - احتمال اینکه وقتی به سیستم نیاز داریم، سیستم در حالت عملیاتی باشد

$$\frac{\text{Mean Time to Failure}}{\text{Mean Time to Failure + Mean Time to Repair}}$$

سناریوی عینی قابلیت دسترسی

Source:
External to System

Stimulus:
Unanticipated Message

Environment:
Normal Operation

Response:
Inform Operator Continue to Operate

Response Measure:
No Downtime

قابلیت اصلاح‌پذیری

- قابلیت اصلاح‌پذیری دربارهٔ هزینهٔ تغییرات است
- این قابلیت دو نگرانی بوجود می‌آورد

۱- چه چیزی می‌تواند تغییر کند (فرآورده)؟

تغییر می‌تواند در هر یک از جنبه‌های سیستم مانند:

- وظایفی که سیستم انجام می‌دهد
- سکویی که سیستم روی آن قرار دارد (سخت‌افزار، سیستم‌عامل و ...)
- محیطی که سیستم در آن اجرا می‌شود
- خصوصیات کیفیتی که سیستم نمایش می‌دهد

قابلیت اصلاح‌پذیری (ادامه)

۲- چه وقت تغییر انجام شود و چه کسی تغییر را انجام دهد (محیط)؟

- **تمام این عملیات زمان و بودجه می‌خواهند که قابل اندازه‌گیری است**

سناریوی عمومی قابلیت اصلاح‌پذیری

Portion of Scenario	Possible Values
Source	End user, developer, system administrator
Stimulus	Wishes to add/delete/modify/vary functionality, quality attribute, capacity
Artifact	System user interface, platform, environment; system that interoperates with target system
Environment	At runtime, compile time, build time, design time
Response	Locates places in architecture to be modified; makes modification without affecting other functionality; tests modification; deploys modification
Response Measure	Cost in terms of number of elements affected, effort, money; extent to which this affects other functions or quality attributes

سناریوی عینی قابلیت اصلاح‌پذیری

Source:
Developer

Stimulus:
Wishes to Change the UI

Environment:
At Design Time

Response:
Modification Is Made with No Side Effects

Response Measure:
In Three Hours

کارایی

- کارایی در مورد زمانبندی است
  - وقفه‌ها، پیام‌ها، درخواست‌های کاربران، یا مدت زمان
  - وقتی رویدادی اتفاق می‌افتد چه مدت طول می‌کشد تا سیستم پاسخ دهد
- پیچیدگی
  - تعداد منابع رویداد و ترتیب اتفاق افتادن آنها
  - این خصوصیت، زبانی برای ساخت سناریوهای کارایی عمومی است

سناریوهای کارایی

- اغلب سناریوهای کارایی
  - با درخواست برای یک سرویس در سیستم شروع می‌شوند
  - برآورده نمودن درخواست با مصرف منابع همراه است
  - معمولاً رویدادها به صورت موازی بکار می‌روند
- الگوهای رسیدن رویدادها (*Arrival Patterns*)
  - دوره‌ای: اغلب در سیستم‌های آنی دیده می‌شوند
  - احتمالی: رویدادها بر طبق توزیع احتمال خاصی دریافت می‌شوند
  - منفرد: الگویی که نمی‌تواند با دیگر الگوها نمایش داده شود

سناریوی عمومی کارایی

Portion of Scenario	Possible Values
Source	One of a number of independent sources, possibly from within system
Stimulus	Periodic events arrive; sporadic events arrive; stochastic events arrive
Artifact	System
Environment	Normal mode; overload mode
Response	Processes stimuli; changes level of service
Response Measure	Latency, deadline, throughput, jitter, miss rate, data loss

نمونه سناریوهای کارایی

Source:
Users

Stimulus:
Initiate Transactions

Environment:
Under Normal Operations

Response:
Transactions Are Processed

Response Measure:
With Average Latency of Two Seconds

امنیت

- امنیت معیاری برای توانایی سیستم برای مقاومت در مقابل استفاده غیرمجاز است در حالیکه سیستم خدمات خود را برای کاربران مجاز ارائه می‌دهد
- هر تلاشی برای شکستن امنیت، حمله نامیده می‌شود
  - حمله می‌تواند برای دستیابی غیرمجاز به داده‌ها یا خدمات، تغییر آنها، یا با هدف از کار انداختن سرویس‌دهنده برای کاربران مجاز باشد
- امنیت می‌تواند به عنوان صفتی برای سیستمی که فراهم‌کنندهٔ اعتبار، جامعیت، تضمین، قابلیت دسترسی و ثبت وقایع است بیان شود

سناریوی عمومی امنیت

Portion of Scenario	Possible Values
Source	Individual or system that is correctly identified, identified incorrectly, of unknown identity who is internal/external, authorized/not authorized with access to limited resources, vast resources
Stimulus	Tries to display data, change/delete data, access system services, reduce availability to system services
Artifact	System services; data within system
Environment	Either online or offline, connected or disconnected, firewalled or open
Response	Authenticates user; hides identity of the user; blocks access to data and/or services; allows access to data and/or services; grants or withdraws permission to access data and/or services; records access/modifications or attempts to access/modify data/services by identity;
Response Measure	Time/effort/resources required to circumvent security measures with probability of success; probability of detecting attack; probability of identifying individual responsible for attack or access/modification of data and/or services;

سناریوی امنیت نمونه

Source:
Correctly Identified Individual

Stimulus:
Tries to Modify Information

Artifact
Data within the System

Environment:
Under Normal Operations

Response:
System Maintains Audit Trial

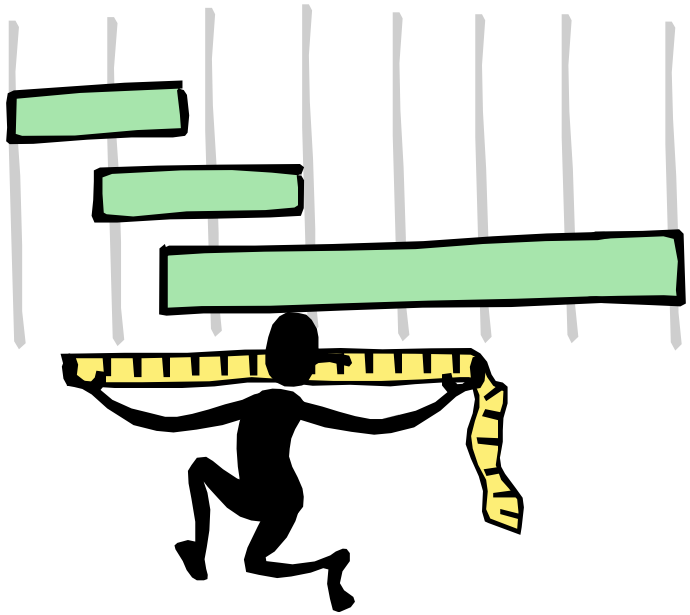
Response Measure:
Correct Data Is Restored within a Day

قابلیت تست

- به میزان راحتی که نرم‌افزار می‌تواند خطاهای خود را در هنگام تست نمایش دهد، عنوان می‌شود
- حداقل ۴۰ درصد از هزینه توسعه سیستم‌های خوب مهندسی شده در هنگام تست است
  - اگر معمار بتواند این هزینه را کاهش دهد، بازدهی بالا خواهد بود
- در عمل، قابلیت تست، به احتمال اینکه نرم‌افزار حداقل دارای یک خطا باشد که در اجرای بعدی بروز خواهد نمود، اشاره دارد
  - ترکیب معیارهای متفاوت سبب پیچیده شدن این محاسبه می‌شود

قابلیت تست (ادامه)

- برای اینکه یک سیستم بدرستی قابل تست باشد، باید بتوان حالت داخلی و ورودی‌های هر مولفه را کنترل نموده و سپس خروجی را مشاهده نمود



سناریوی عمومی قابلیت تست

Portion of Scenario	Possible Values
Source	Unit developer, Increment integrator, System Verifier, Client acceptance tester, System user
Stimulus	Analysis, architecture, design, class, subsystem integration completed; system delivered
Artifact	Piece of design, piece of code, complete application
Environment	At design time, at development time, at compile time, at deployment time
Response	Provides access to state values; provides computed values; prepares test environment
Response Measure	Percent executable statements executed Probability of failure if fault exists Time to perform tests Length of longest dependency chain in a test Length of time to prepare test environment

نمونه سناریوی قابلیت تست

Source:
Unit Tester

Stimulus:

Performs Unit Test

Environment:

At the Completion of the Component

Response:

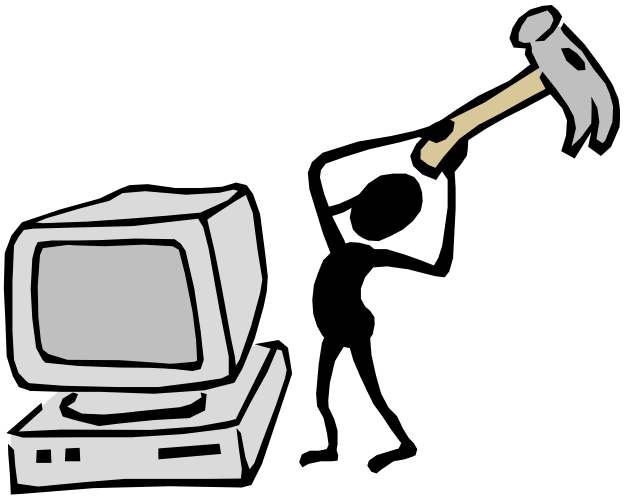
Component has Interface for Controlling Behavior and Output of the Component is Observable

Response Measure:

Path coverage of 85% is achieved within Three Hours

قابلیت استفاده

- چقدر برای یک کاربر انجام کار دلخواهش آسان است
- سیستم از کاربران چه حمایت‌هایی می‌نماید
  - یادگیری خصوصیات سیستم
  - استفاده از کارایی سیستم
  - کاهش تاثیرات خطا
  - تطبیق سیستم با نیازهای کاربران
  - افزایش اطمینان و رضایت



سناریوی عمومی قابلیت استفاده

Portion of Scenario	Possible Values
Source	End user
Stimulus	Wants to learn system features; use system efficiently; minimize impact of errors; adapt system; feel comfortable
Artifact	System
Environment	At runtime or configure time
Response	System provides one or more of the following responses: to support "learn system features" , to support "use system efficiently", "minimize impact of errors "
Response Measure	Task time, number of errors, number of problems solved, user satisfaction, gain of user knowledge, ratio of successful operations to total operations, amount of time/data lost

نمونه سناریوی قابلیت استفاده

Source:
Users

Stimulus:
Minimize impact of Errors

Artifact
System

Environment:
At Runtime

Response:
Wishes to Cancel Current Operations

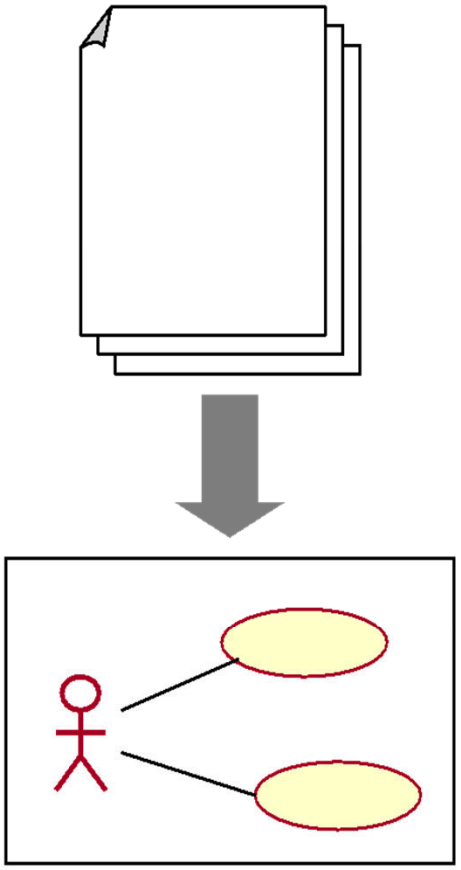
Response Measure:
Cancellation Takes less than One Second

نکاتی در مورد سناریوها

- نقش سناریوهای عینی (*Concrete*) برای نیازمندی‌های خصوصیات کیفیتی همانند نقش موارد کاربری برای نیازمندی‌های وظیفه‌مندی است
- مجموعه‌ای از سناریوهای عینی می‌توانند برای نمایش نیازمندی‌های خصوصیات کیفیتی برای یک سیستم مورد استفاده قرار گیرند
- یکی از موارد استفاده از سناریوهای عمومی برقراری ارتباط بین ذینفعان است

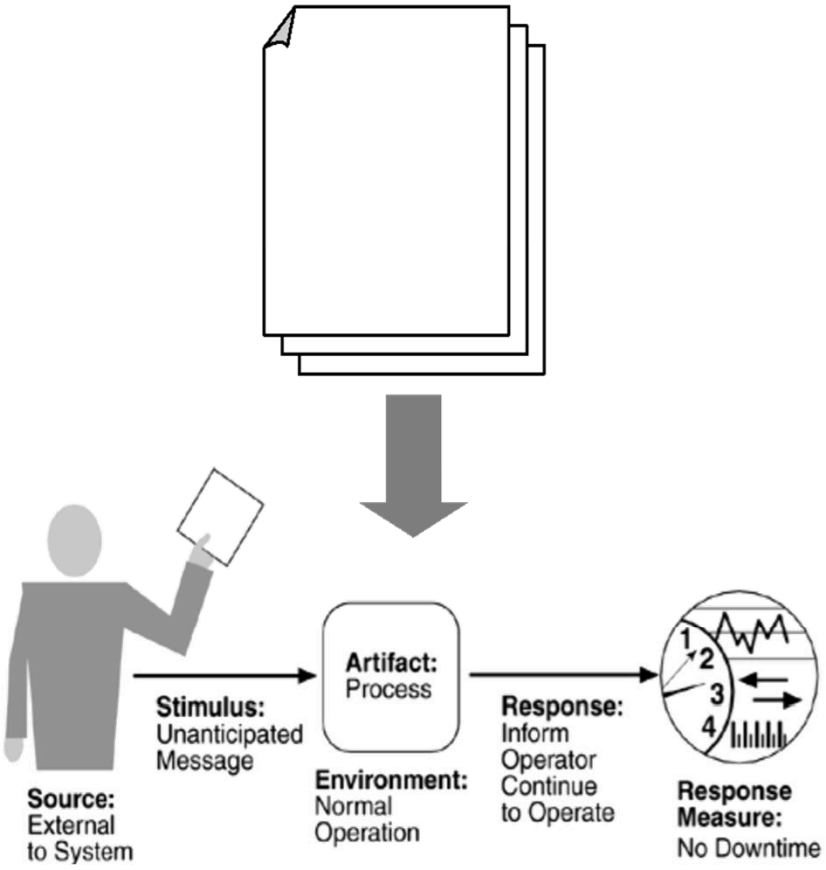
نکاتی در مورد سناریوها (ادامه)

Functional Requirements



Use Cases

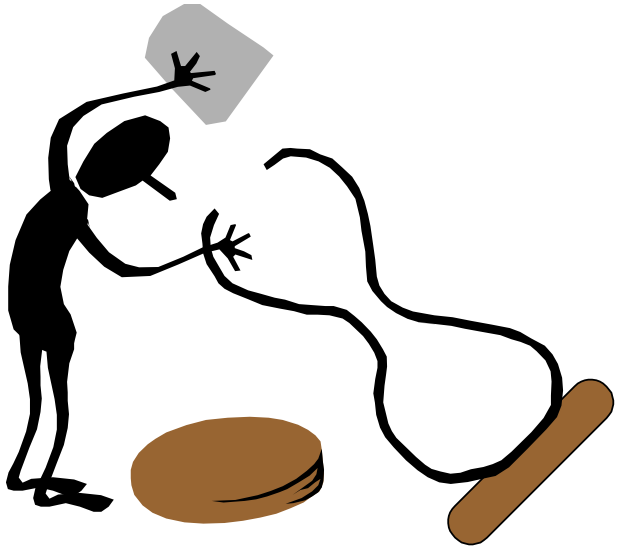
Quality Attributes Requirements



Concrete Scenarios

خصوصیات کیفیتی حرفه

- زمان ارائه به بازار
- هزینه و فایده
- عمر تخمینی سیستم
- بازار هدف
- زمان‌بندی معرفی به بازار
- مجتمع سازی با سیستم‌های موروثی



خصوصیات کیفیتی معماری

- **جامعیت مفهومی**
  - سبب یکپارچگی تمام سطوح سیستم می‌شود
  - معماری باید کارهای یکسان را به روش‌های یکسان انجام دهد
  - یکی از مهمترین نکات در طراحی سیستم است
- **صحت و کامل بودن**
  - اجازه می‌دهد تمام نیازمندی‌ها و اجبارهای منابع در زمان اجرا برآورده شوند
  - برای معماری حیاتی است

خصوصیات کیفیتی معماری (ادامه)

قابلیت ساخت

- اجازه می‌دهد سیستم توسط تیم موجود به موقع کامل شود
- اجازه می‌دهد سیستم برای تغییرات بعدی در حین روند توسعه باز باشد
- به سادگی ساخت سیستم اشاره دارد