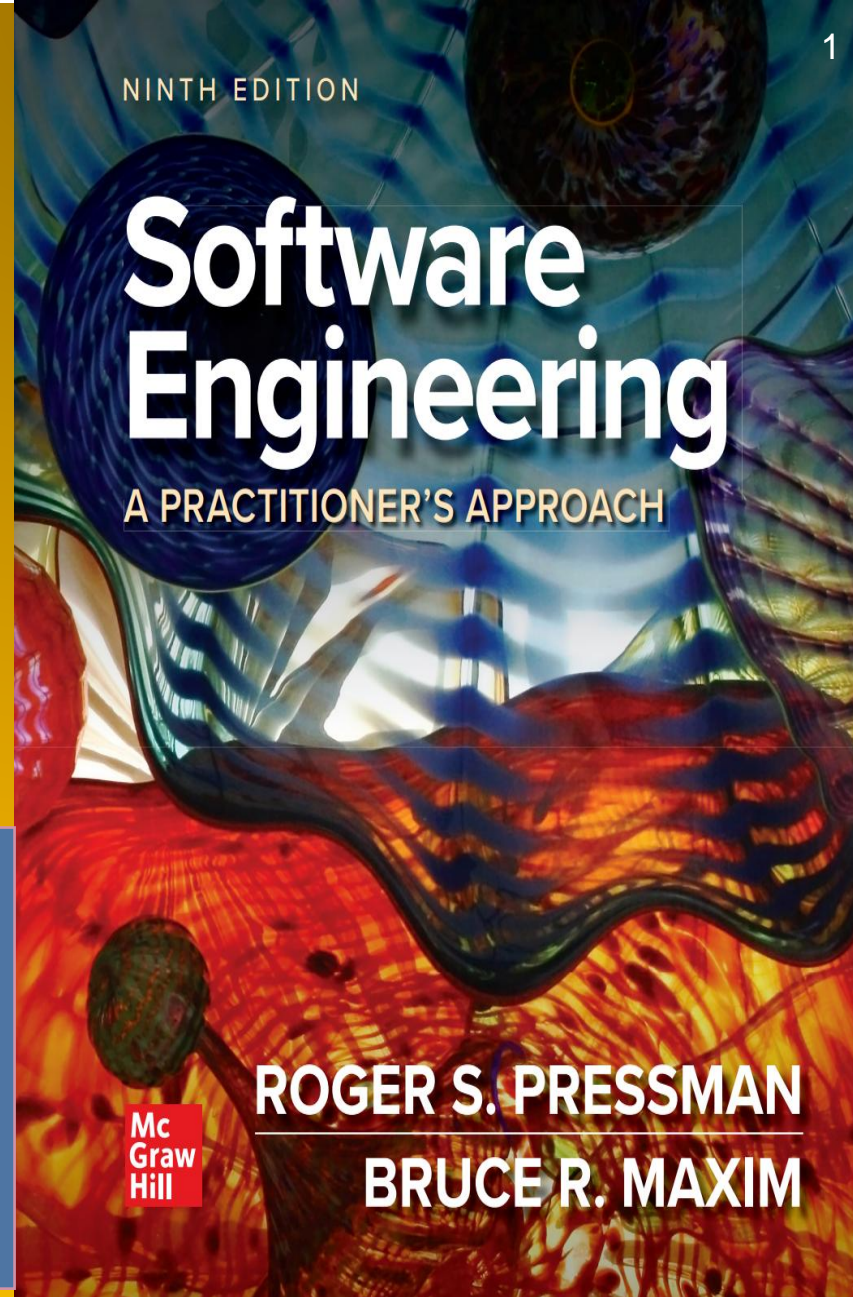


# Chapter 4 – Requirements Engineering

Part 1



## Topics covered

- الزامات کارکردی و غیر کارکردی
- سند الزامات نرم‌افزاری
- مشخصات الزامات
- فرآیندهای مهندسی الزامات
- استخراج و تحلیل الزامات
- اعتبارسنجی الزامات
- مدیریت الزامات

## مهندسی الزامات Requirements engineering

- فرآیند تعیین خدماتی که مشتری از یک سیستم نیاز دارد و محدودیت‌هایی که سیستم تحت آنها کار می‌کند و توسعه می‌یابد
- خود الزامات شامل توصیفاتی از خدمات سیستم و محدودیت‌هایی هستند که در طول فرآیند مهندسی الزامات تولید می‌شوند.

## الزامات چیست؟

- الزامات ممکن است از یک بیانیه انتزاعی سطح بالا درباره یک خدمت یا یک محدودیت سیستمی، تا یک مشخصات عملکردی ریاضی دقیق متغیر باشند.
- این امر اجتناب‌ناپذیر است زیرا الزامات می‌توانند دو نقش داشته باشند:
  - ممکن است مبنایی برای پیشنهاد مناقصه یک قرارداد باشند - بنابراین باید به گونه‌ای بیان شوند که قابل تفسیر باشند؛
  - ممکن است مبنای خود قرارداد باشند - بنابراین باید با جزئیات تعریف شوند؛
- هر دوی این بیانیه‌ها ممکن است به عنوان الزامات شناخته شوند.

# What is a requirement?

- ۱. بیان کلی نیازمندی (به عنوان مبنای مناقصه)
- نیازمندی "سیستم باید امکان ذخیره‌سازی امن داده‌های مشتری را فراهم کند".
- تفسیر: این نیازمندی کلی و مبهم است و به پیمانکاران اجازه می‌دهد که راه‌حل‌های مختلفی برای امنیت ذخیره‌سازی ارائه دهند (مثلاً رمزنگاری، بلاک‌چین، یا الگوریتم‌های اختصاصی).
- ۲. محدودیت کلی سیستم (به عنوان مبنای مناقصه)
- نیازمندی "برنامه باید از دسترس‌پذیری بالا پشتیبانی کند".
- تفسیر: پیمانکاران می‌توانند روش‌های متفاوتی برای دسترس‌پذیری بالا پیشنهاد دهند (مانند توزیع بار یا پشتیبانی از سیستم‌های ابری).
- ۳. مشخصات دقیق ریاضی عملکرد (به عنوان مبنای قرارداد)
- نیازمندی "سیستم باید توانایی پاسخگویی به ۱۰ هزار کاربر هم‌زمان را با زمان پاسخگویی حداکثر ۲۰۰ میلی‌ثانیه و دسترس‌پذیری ۹۹,۹٪ در هر دوره ۳۰ روزه داشته باشد".
- توضیح: این نیازمندی دقیق و قابل اندازه‌گیری است و به عنوان معیار عملکرد در قرارداد استفاده می‌شود.
- ۴. محدودیت دقیق سیستم (به عنوان مبنای قرارداد)
- نیازمندی "داده‌های سیستم باید با استاندارد AES-256 در حین انتقال و ذخیره‌سازی رمزنگاری شوند".
- توضیح: این نیازمندی مشخص است و انتخاب پروتکل امنیتی را محدود می‌کند؛ در نتیجه، برای قرارداد مناسب است که تعهدات را شفاف‌تر می‌کند.

## Requirements abstraction (Davis)

- اگر شرکتی بخواهد قراردادی برای یک پروژه بزرگ توسعه نرم افزار منعقد کند، باید نیازهای خود را به شکلی به اندازه کافی انتزاعی تعریف کند تا یک راه حل از پیش تعیین نشود.
- الزامات باید به گونه ای نوشته شوند که چندین پیمانکار بتوانند برای قرارداد پیشنهاد بدهند و شاید روش های مختلفی را برای برآوردن نیازهای سازمان مشتری ارائه کنند.
- پس از اعطای قرارداد، پیمانکار باید تعریف سیستمی را با جزئیات بیشتر برای مشتری بنویسد تا مشتری بتواند درک کند و تأیید کند که نرم افزار چه عملکردی خواهد داشت. هر دوی این اسناد ممکن است به عنوان سند الزامات برای سیستم نامیده شوند."

## انواع الزامات

### • الزامات کاربر: توضیح سطح بالا و قابل فهم برای کاربر نهایی

- بیانیه‌هایی به زبان طبیعی به همراه نمودارهایی از خدماتی که سیستم ارائه می‌دهد و محدودیت‌های عملیاتی آن. برای مشتریان نوشته می‌شود.

### • الزامات سیستم: مشخصه‌های فنی و دقیق برای هدایت برنامه

### نویسان

- سندی ساخت یافته که توصیفات دقیقی از عملکردها، خدمات و محدودیت‌های عملیاتی سیستم را بیان می‌کند. مشخص می‌کند چه چیزی باید پیاده‌سازی شود و ممکن است بخشی از قرارداد بین مشتری و پیمانکار باشد.



# الزامات کاربر و سیستم

## تعریف الزامات کاربر

1. سیستم MHC-PMS باید گزارش‌های مدیریتی ماهانه‌ای تولید کند که هزینه داروهای تجویز شده توسط هر کلینیک را در آن ماه نشان دهد.

## مشخصات الزامات سیستم

- 1.1 در آخرین روز کاری هر ماه، خلاصه‌ای از داروهای تجویز شده، هزینه آن‌ها و کلینیک‌های تجویزشان باید تولید شود.
- 1.2 سیستم باید به طور خودکار گزارش را برای چاپ بعد از ساعت ۱۷:۳۰ در آخرین روز کاری ماه تولید کند.
- 1.3 برای هر کلینیک یک گزارش ایجاد شود که نام‌های تک‌تک داروها، تعداد کل نسخه‌ها، تعداد دوزهای تجویز شده و کل هزینه داروهای تجویز شده را فهرست کند.
- 1.4 اگر داروها در واحدهای دوز مختلف (مثلاً ۱۰ میلی‌گرم، ۲۰ میلی‌گرم، و غیره) موجود باشند، گزارش‌های جداگانه‌ای برای هر واحد دوز ایجاد شود.
- 1.5 دسترسی به تمام گزارش‌های هزینه محدود به کاربران مجاز باشد که در فهرست کنترل دسترسی مدیریتی ذکر شده‌اند.



# خوانندگان انواع مختلف مشخصات الزامات

## الزامات سیستم

- کاربران نهایی سیستم
- مهندسان مشتری
- معماران سیستم
- توسعه دهندگان نرم افزار

## الزامات کاربر

- مدیران مشتری
- کاربران نهایی سیستم
- مهندسان مشتری
- مدیران پیمانکار
- معماران سیستم

# Functional and non-functional requirements

## • الزامات کارکردی **Functional**

- بیانیه‌هایی از خدماتی که سیستم باید ارائه دهد، نحوه واکنش سیستم به ورودی‌های خاص و رفتار سیستم در شرایط خاص.
- ممکن است شامل مواردی باشد که سیستم نباید انجام دهد.

## • الزامات غیر کارکردی

- محدودیت‌هایی بر خدمات یا عملکردهای ارائه‌شده توسط سیستم، مانند محدودیت‌های زمانی، محدودیت‌های فرآیند توسعه، استانداردها و غیره.
- غالباً بر سیستم به‌طور کلی اعمال می‌شوند، نه بر ویژگی‌ها یا خدمات خاص.

## • الزامات حوزه **Domain**

- محدودیت‌هایی بر سیستم که از حوزه عملیات سیستم نشأت می‌گیرند.

## Functional requirements for the MHC-PMS

### • نمونه‌هایی از الزامات کارکردی

- کاربر باید بتواند فهرست وقت‌های ملاقات تمام کلینیک‌ها را جستجو کند.
- سیستم باید هر روز، برای هر کلینیک، فهرستی از بیماران که انتظار می‌رود آن روز به ملاقات‌ها مراجعه کنند، ایجاد کند.
- هر کارمند استفاده‌کننده از سیستم باید به‌طور یکتا با شماره کارمندی ۸ رقمی خود شناسایی شود.

## Requirements imprecision

- **مسائل ناشی از الزامات مبهم**
  - مشکلاتی ایجاد می‌شوند وقتی الزامات به‌طور دقیق بیان نشده‌اند.
  - الزامات مبهم ممکن است توسط توسعه‌دهندگان و کاربران به روش‌های مختلفی تفسیر شوند.
  - به عنوان مثال، اصطلاح "جستجو" در الزام ۱:
    - **قصد کاربر** - جستجو برای نام بیمار در بین تمام وقت‌های ملاقات در تمام کلینیک‌ها؛
    - **تفسیر توسعه‌دهنده** - جستجو برای نام بیمار در یک کلینیک خاص.کاربر کلینیک را انتخاب کرده و سپس جستجو می‌کند.

## Requirements completeness and consistency

### • اصول تکمیل و سازگاری الزامات

• **کامل بودن:** الزامات باید شامل توصیف همه امکانات موردنیاز باشند.

• **سازگاری:** نباید تناقضی در توصیفات امکانات سیستم وجود داشته باشد.

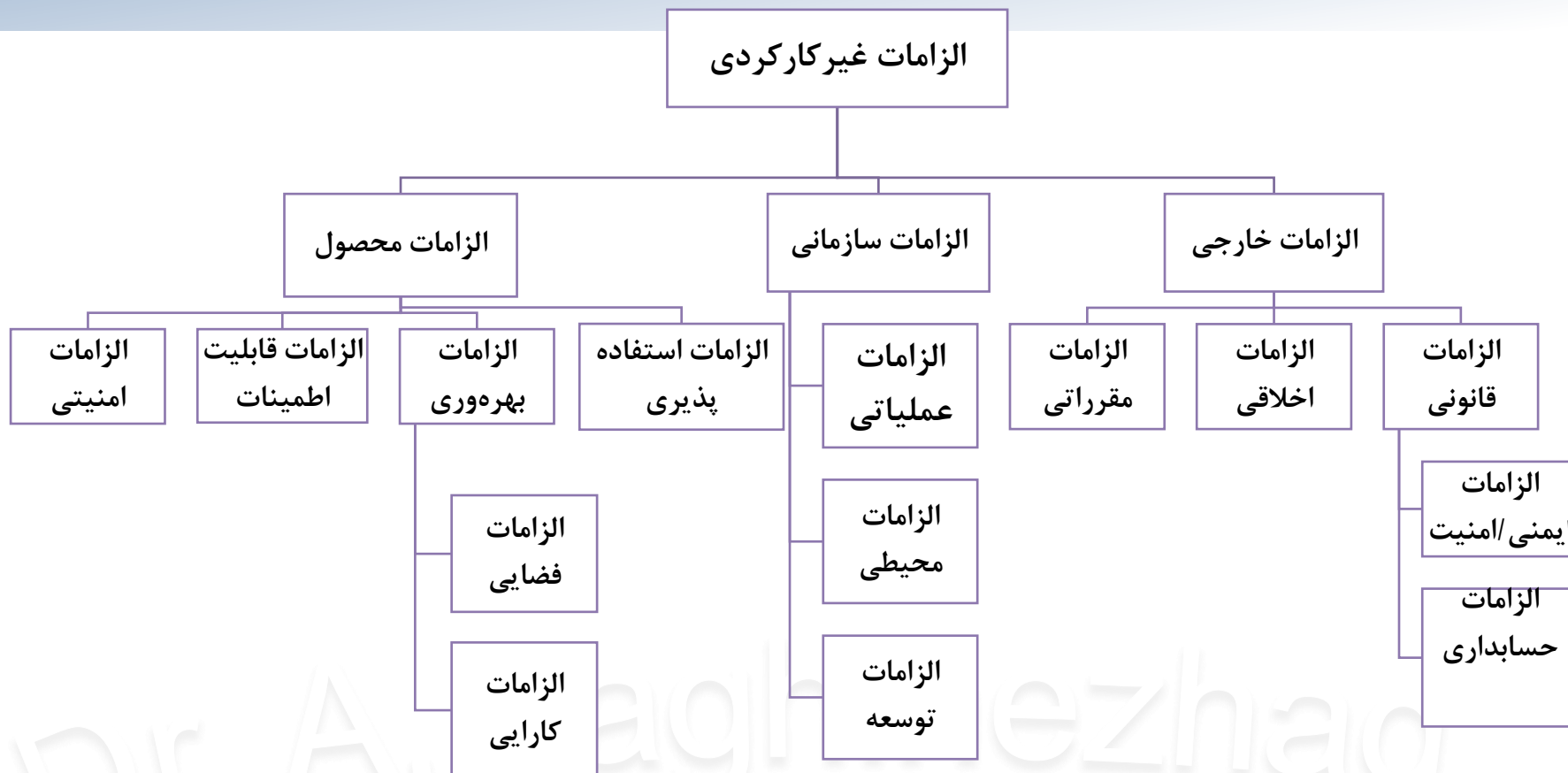
• در عمل، تهیه یک سند الزامات کاملاً کامل و سازگار غیرممکن است.

## الزامات غیر کار کردی

### • الزامات غیر کار کردی

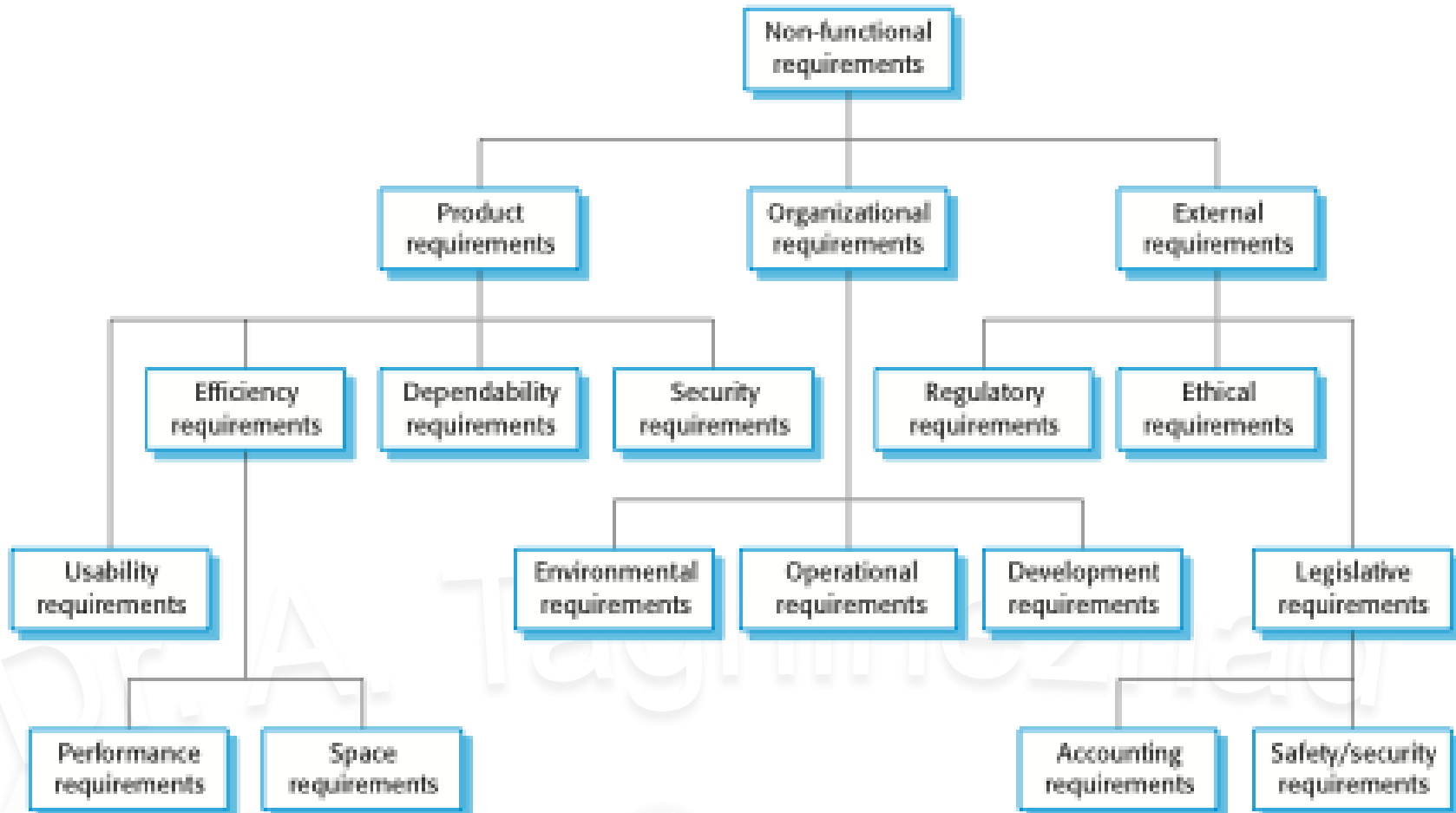
- ویژگی‌ها و محدودیت‌های سیستم مانند قابلیت اطمینان، زمان پاسخ و نیازهای ذخیره‌سازی را تعریف می‌کنند. محدودیت‌ها شامل قابلیت دستگاه‌های ورودی/خروجی، نمایش‌های سیستمی و غیره می‌شوند.
- ممکن است نیازهای فرآیندی را نیز مشخص کنند که استفاده از محیط توسعه خاص، زبان برنامه‌نویسی یا روش توسعه خاصی را اجباری می‌کنند.
- الزامات غیر کار کردی ممکن است از الزامات کار کردی مهم‌تر باشند؛ اگر این الزامات برآورده نشوند، سیستم ممکن است غیر قابل استفاده شود.

# Types of nonfunctional requirement





# Types of nonfunctional requirement



## Non-functional requirements implementation

- **نیازمندی‌های غیرعملکردی** می‌توانند معماری کلی سیستم را تحت تأثیر قرار دهند، نه فقط اجزای خاص.

□ **مثال:** اگر نیازمندی غیرعملکردی شما امنیت باشد، ممکن است سیستم را طوری طراحی کنید که همه داده‌های ارسال شده بین اجزا رمزگذاری شوند. این نیازمندی امنیتی می‌تواند نیازمندی‌های عملکردی جدیدی مانند پیاده‌سازی سرویس رمزنگاری یا محدود کردن دسترسی کاربران به بخش‌های خاصی از سیستم را ایجاد کند. همچنین ممکن است قوانین و محدودیت‌هایی برای نیازمندی‌های موجود، مانند نحوه احراز هویت کاربران، اعمال کند.

## طبقه بندی های غیر کاربردی

- نیازمندی های محصول
  - نیازمندی هایی که مشخص می کنند محصول ارائه شده باید به روشی خاص عمل کند، مانند سرعت اجرا، قابلیت اطمینان و غیره.
- نیازمندی های سازمانی
  - نیازمندی هایی که نتیجه سیاست ها و رویه های سازمانی هستند، مانند استانداردهای فرآیند استفاده شده، نیازمندی های پیاده سازی و غیره.
- نیازمندی های خارجی
  - نیازمندی هایی که از عواملی که خارج از سیستم و فرآیند توسعه آن هستند ناشی می شوند، مانند نیازمندی های تعامل پذیری، نیازمندی های قانونی و غیره.

## نمونه هایی از الزامات غیر کاربردی در MHC-PMS

### نیازمندی محصول

سیستم MHC-PMS باید در ساعات کاری معمول (دوشنبه تا جمعه، ۰۸:۳۰ تا ۱۷:۳۰) در دسترس تمام کلینیک‌ها باشد. زمان خاموشی در ساعات کاری معمول نباید در هر روز بیشتر از پنج ثانیه باشد.

### نیازمندی سازمانی

کاربران سیستم MHC-PMS باید با استفاده از کارت شناسایی خود احراز هویت کنند.

### نیازمندی خارجی

سیستم باید الزامات حریم خصوصی بیمار را طبق HStan-03-2006-priv پیاده‌سازی کند.

## اهداف و الزامات

- بیان دقیق نیازمندی‌های غیرعملکردی ممکن است بسیار دشوار باشد و نیازمندی‌های مبهم ممکن است سخت به تأیید برسند
- هدف
  - نیاز عمومی کاربر مانند آسانی استفاده.
- نیازمندی غیرعملکردی قابل تأیید
  - بیانی با استفاده از برخی اندازه‌گیری‌ها که می‌تواند به طور عینی آزمایش شود.
- اهداف برای توسعه‌دهندگان مفید هستند زیرا نیات کاربران سیستم را منتقل می‌کنند

## الزامات قابلیت استفاده

- سیستم باید به راحتی توسط کادر پزشکی قابل استفاده باشد و باید به گونه‌ای سازماندهی شود که خطاهای کاربر به حداقل برسد. (هدف)
- کادر پزشکی باید بتوانند پس از چهار ساعت آموزش، از تمام عملکردهای سیستم استفاده کنند. پس از این آموزش، میانگین تعداد خطاهای انجام شده توسط کاربران با تجربه نباید در هر ساعت استفاده از سیستم بیشتر از دو خطا باشد. (نیازمندی غیرعملکردی قابل آزمایش)

ویژگی‌ها	معیارهای اندازه‌گیری
سرعت	<ul style="list-style-type: none"> <li>• تعداد تراکنش‌های پردازش شده در ثانیه</li> <li>• زمان پاسخگویی به کاربر/ارویداد</li> <li>• زمان تازه‌سازی صفحه</li> </ul>
اندازه	<p>Mbytes Number of ROM chips</p>
سهولت استفاده	<ul style="list-style-type: none"> <li>• زمان آموزش</li> <li>• تعداد صفحات راهنما</li> </ul>
قابلیت اتکا یا اطمینان	<p>Mean time to failure Probability of unavailability Rate of failure occurrence Availability</p>
استحکام	<ul style="list-style-type: none"> <li>• زمان لازم برای راه‌اندازی مجدد پس از خرابی</li> <li>• درصد رویدادهایی که موجب خرابی می‌شوند</li> <li>• احتمال خرابی داده‌ها در صورت وقوع خرابی</li> </ul>
قابلیت حمل	<ul style="list-style-type: none"> <li>• درصد دستورات وابسته به سیستم هدف</li> <li>• تعداد سیستم‌های هدف</li> </ul>



# Metrics for specifying nonfunctional requirements

Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Mbytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

## مسائل الزامات دامنه (حوزه)

- دامنه عملیاتی سیستم، نیازمندی‌هایی را بر سیستم تحمیل می‌کند.

- به عنوان مثال، یک سیستم کنترل قطار باید ویژگی‌های ترمز در شرایط آب و هوایی مختلف را در نظر بگیرد.

- نیازمندی‌های دامنه می‌توانند نیازمندی‌های عملکردی جدید، محدودیت‌هایی بر نیازمندی‌های موجود یا محاسبات خاصی را تعریف کنند.

- اگر نیازمندی‌های دامنه برآورده نشوند، سیستم ممکن است غیرقابل کار باشد .

## سیستم حفاظت از قطار

• این یک نیازمندی حوزه برای سیستم حفاظت از قطار است:  
کاهش سرعت قطار باید به صورت زیر محاسبه شود:

$$\square \square D_{\text{train}} = D_{\text{control}} + D_{\text{gradient}}$$

□ که در آن  $D_{\text{gradient}}$  برابر است با  $۹.۸۱ * \text{ms}^2$  شیب جبرانی/آلفا و جایی که مقادیر  $\text{ms}^2/۹.۸۱$  آلفا برای انواع مختلف قطار شناخته شده‌اند.

درک این موضوع و نحوه تعامل آن با سایر نیازمندی‌ها برای یک غیرمتخصص دشوار است.

## مسائل الزامات دامنه (حوزه)

- قابل فهم بودن
  - نیازمندی‌ها به زبان حوزه کاربرد بیان می‌شوند؛
  - اغلب توسط مهندسان نرم‌افزار که سیستم را توسعه می‌دهند، درک نمی‌شود.
- ذاتی بودن
  - متخصصان حوزه آنقدر به این حوزه آگاه هستند که به فکر صریح کردن نیازمندی‌های حوزه نمی‌افتند.

## نکات کلیدی

- نیازمندی‌های یک سیستم نرم‌افزاری آنچه سیستم باید انجام دهد را مشخص کرده و محدودیت‌هایی را برای عملکرد و پیاده‌سازی آن تعریف می‌کند .
- نیازمندی‌های عملکردی، بیانیه‌هایی از خدماتی هستند که سیستم باید ارائه دهد یا توصیفاتی از نحوه انجام برخی محاسبات هستند .
- نیازمندی‌های غیرعملکردی معمولاً سیستم در حال توسعه و فرآیند توسعه مورد استفاده را محدود می‌کند .
- این نیازمندی‌ها اغلب به ویژگی‌های نوظهور سیستم مرتبط می‌شوند و بنابراین به کل سیستم اعمال می‌شوند .

NINTH EDITION

# Software Engineering

A PRACTITIONER'S APPROACH

ROGER S. PRESSMAN

BRUCE R. MAXIM

Mc  
Graw  
Hill

## Chapter 4 – Requirements Engineering

Part 2

## اسناد مورد نیاز نرم افزار

- سند نیازمندی‌های نرم‌افزاری، بیانیه رسمی از آنچه از توسعه‌دهندگان سیستم خواسته می‌شود است
- باید شامل هر دو تعریف نیازمندی‌های کاربر و مشخصات نیازمندی‌های سیستم باشد.
- این سند یک سند طراحی نیست. تا حد امکان، باید آنچه سیستم باید انجام دهد (WHAT) را تعیین کند نه اینکه چگونه باید آن را انجام دهد. (HOW).



## روش ها و الزامات چابک

- بسیاری از روش‌های چابک (Agile) معتقدند که تهیه یک سند نیازمندی‌ها هدر دادن وقت است زیرا نیازمندی‌ها به سرعت تغییر می‌کنند .
  - بنابراین، این سند همیشه قدیمی و از تاریخ گذشته است .
  - روش‌هایی مانند XP از مهندسی نیازمندی‌های تدریجی استفاده می‌کنند و نیازمندی‌ها را به صورت «داستان‌های کاربر» بیان می‌کنند (که در فصل ۳ بحث شده است) .
  - این برای سیستم‌های تجاری عملی است اما برای سیستم‌هایی که به تحلیل‌های پیش‌تحویلی زیادی نیاز دارند (مانند سیستم‌های بحرانی) یا سیستم‌هایی که توسط چندین تیم توسعه داده می‌شوند، مشکل‌ساز است .

# کاربران یک سند الزامات

## مشتریان سیستم

- نیازمندی‌ها را مشخص کرده و آنها را می‌خوانند تا بررسی کنند که آیا نیازهایشان را برآورده می‌کنند یا خیر. مشتریان تغییرات را در نیازمندی‌ها مشخص می‌کنند.

## مدیران

- از سند نیازمندی‌ها برای برنامه‌ریزی یک پیشنهاد برای سیستم و برنامه‌ریزی فرآیند توسعه سیستم استفاده می‌کنند .

## مهندسان سیستم

- از نیازمندی‌ها برای درک آنچه سیستم قرار است توسعه یابد، استفاده می‌کنند .

## مهندسان آزمایش سیستم

- از نیازمندی‌ها برای توسعه آزمایش‌های اعتبارسنجی برای سیستم استفاده می‌کنند .

## مهندسان نگهداری سیستم

- از نیازمندی‌ها برای درک سیستم و روابط بین اجزای آن استفاده می‌کنند .

## Requirements document variability: تغییرپذیری سند نیازمندی‌ها

- اطلاعات موجود در سند نیازمندی‌ها به نوع سیستم و رویکرد توسعه‌ای که استفاده می‌شود بستگی دارد .
- سیستم‌هایی که به طور تدریجی توسعه داده می‌شوند معمولاً جزئیات کمتری در سند نیازمندی‌ها دارند .
- استانداردهای اسناد نیازمندی‌ها طراحی شده‌اند، مانند استاندارد IEEE. این استانداردها بیشتر به نیازمندی‌های پروژه‌های مهندسی سیستم‌های بزرگ قابل اعمال هستند .

Dr. A. Taghinezhad

# ساختار یک سند الزامات

فصل	توضیحات
پیش‌گفتار	این باید خوانندگان مورد انتظار سند را تعریف کرده و تاریخچه نسخه آن را توصیف کند، از جمله دلایل ایجاد نسخه جدید و خلاصه‌ای از تغییرات انجام شده در هر نسخه .
مقدمه	این باید نیاز سیستم را توصیف کند. باید به طور مختصر عملکردهای سیستم را توضیح دهد و بیان کند که چگونه با سایر سیستم‌ها کار خواهد کرد. همچنین باید توضیح دهد که چگونه سیستم در اهداف کلی کسب‌وکار یا استراتژیک سازمانی که نرم‌افزار را سفارش داده است، قرار می‌گیرد .
فهرست واژگان	این باید اصطلاحات فنی استفاده شده در سند را تعریف کند. نباید فرضیاتی درباره تجربه یا تخصص خواننده داشته باشید .
تعریف نیازمندی‌های کاربر	در اینجا، شما خدمات ارائه شده برای کاربر را توصیف می‌کنید. نیازمندی‌های غیرعملکردی سیستم نیز باید در این بخش توصیف شوند. این توصیف ممکن است از زبان طبیعی، نمودارها یا سایر نشانه‌ها که برای مشتریان قابل درک است، استفاده کند. استانداردهای محصول و فرآیند که باید رعایت شوند، باید مشخص شوند .
معماری سیستم	این فصل باید یک نمای کلی از معماری مورد انتظار سیستم را ارائه دهد و توزیع عملکردها در میان ماژول‌های سیستم را نشان دهد. اجزای معماری که مجدداً استفاده می‌شوند باید هایلایت شوند .

# The structure of a requirements document

Chapter	Description
Preface	This should define the expected readership of the document and describe its version history, including a rationale for the creation of a new version and a summary of the changes made in each version.
Introduction	This should describe the need for the system. It should briefly describe the system's functions and explain how it will work with other systems. It should also describe how the system fits into the overall business or strategic objectives of the organization commissioning the software.
Glossary	This should define the technical terms used in the document. You should not make assumptions about the experience or expertise of the reader.
User requirements definition	Here, you describe the services provided for the user. The nonfunctional system requirements should also be described in this section. This description may use natural language, diagrams, or other notations that are understandable to customers. Product and process standards that must be followed should be specified.
System architecture	This chapter should present a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules. Architectural components that are reused should be highlighted.

# ساختار یک سند الزامات

فصل	توضیحات
مشخصات نیازمندی‌های سیستم	این باید <b>نیازمندی‌های عملکردی و غیرعملکردی</b> را به تفصیل بیشتری توصیف کند. در صورت لزوم، جزئیات بیشتری نیز ممکن است به نیازمندی‌های غیرعملکردی اضافه شود. رابطه‌ها به سیستم‌های دیگر ممکن است تعریف شوند .
مدل‌های سیستم	این ممکن است شامل <b>مدل‌های گرافیکی سیستم</b> باشد که روابط بین اجزای سیستم و سیستم و محیط آن را نشان می‌دهد. نمونه‌هایی از مدل‌های ممکن شامل مدل‌های شیء، مدل‌های جریان داده یا مدل‌های داده معنایی هستند .
تکامل سیستم	این باید <b>فرضیات بنیادی</b> که سیستم بر اساس آن ساخته شده و هرگونه <b>تغییر پیش‌بینی</b> شده به دلیل تکامل <b>سخت‌افزاری، تغییر نیازهای کاربر و غیره</b> را توصیف کند. این بخش برای طراحان سیستم مفید است زیرا ممکن است به آنها کمک کند از تصمیمات طراحی که تغییرات آینده محتمل سیستم را محدود می‌کند، اجتناب کنند .
پیوست‌ها	این باید <b>اطلاعات دقیق و خاصی</b> که به برنامه‌در حال توسعه مرتبط است را ارائه دهد؛ به عنوان مثال، <b>توصیف‌های سخت‌افزاری و پایگاه داده</b> . نیازمندی‌های سخت‌افزاری حداقل و بهینه‌ترین پیکربندی‌ها برای سیستم را تعریف می‌کنند. نیازمندی‌های پایگاه داده سازمان‌دهی منطقی داده‌های استفاده شده توسط سیستم و روابط بین داده‌ها را تعریف می‌کنند .
نمایه	چندین نمایه برای سند ممکن است شامل شود. علاوه بر یک نمایه الفبایی عادی، ممکن است نمایه‌ای از نمودارها، نمایه‌ای از عملکردها و غیره نیز وجود داشته باشد

# The structure of a requirements document

Chapter	Description
System requirements specification	This should describe the functional and nonfunctional requirements in more detail. If necessary, further detail may also be added to the nonfunctional requirements. Interfaces to other systems may be defined.
System models	This might include graphical system models showing the relationships between the system components and the system and its environment. Examples of possible models are object models, data-flow models, or semantic data models.
System evolution	This should describe the fundamental assumptions on which the system is based, and any anticipated changes due to hardware evolution, changing user needs, and so on. This section is useful for system designers as it may help them avoid design decisions that would constrain likely future changes to the system.
Appendices	These should provide detailed, specific information that is related to the application being developed; for example, hardware and database descriptions. Hardware requirements define the minimal and optimal configurations for the system. Database requirements define the logical organization of the data used by the system and the relationships between data.
Index	Several indexes to the document may be included. As well as a normal alphabetic index, there may be an index of diagrams, an index of functions, and so on.



## Requirements specification

- فرآیند نوشتن نیازمندی‌های کاربر و سیستم در یک سند نیازمندی‌ها.

- نیازمندی‌های کاربر باید برای کاربران نهایی و مشتریانی که زمینه فنی ندارند، قابل فهم باشند.

- نیازمندی‌های سیستم، نیازمندی‌های دقیق‌تری هستند و ممکن است شامل اطلاعات فنی بیشتری باشند.

- نیازمندی‌ها ممکن است بخشی از یک قرارداد برای توسعه سیستم باشند؛

□ بنابراین مهم است که این نیازمندی‌ها تا حد امکان کامل باشند.

## روش های نوشتن مشخصات مورد نیاز سیستم

نمادها	توضیح
زبان طبیعی	نیازمندی‌ها با جملات شماره‌گذاری شده به زبان طبیعی نوشته می‌شوند. هر جمله باید یک نیازمندی را بیان کند.
زبان طبیعی ساختاریافته	نیازمندی‌ها به زبان طبیعی در یک فرم یا الگوی استاندارد نوشته می‌شوند. هر فیلد اطلاعاتی در مورد جنبه‌ای از نیازمندی را فراهم می‌کند.
زبان های توضیحات طراحی	این رویکرد از زبانی مشابه زبان‌های برنامه‌نویسی استفاده می‌کند، اما با ویژگی‌های انتزاعی‌تر برای مشخص کردن نیازمندی‌ها از طریق تعریف یک مدل عملیاتی از سیستم. این رویکرد اکنون به ندرت استفاده می‌شود، اگرچه می‌تواند برای مشخصات رابط مفید باشد.
نمادهای گرافیکی	مدل‌های گرافیکی که با حاشیه‌نویسی‌های متنی تکمیل می‌شوند، برای تعریف نیازمندی‌های عملکردی سیستم استفاده می‌شوند؛ نمودارهای کاربرد و توالی UML معمولاً استفاده می‌شوند.
مشخصات ریاضی	نمادها بر اساس مفاهیم ریاضی مانند ماشین‌های حالت متناهی یا مجموعه‌ها هستند. اگرچه این مشخصات بدون ابهام می‌توانند ابهام در سند نیازمندی‌ها را کاهش دهند، بیشتر مشتریان مشخصات رسمی را نمی‌فهمند. آنها نمی‌توانند بررسی کنند که آیا این مشخصات آنچه را که می‌خواهند نمایان می‌سازد و تمایلی به پذیرش آن به عنوان یک قرارداد سیستم ندارند.

# Ways of writing a system requirements specification

Notation	Description
<b>Natural language</b>	The requirements are written using numbered sentences in natural language. Each sentence should express one requirement.
Structured natural language	The requirements are written in natural language on a standard form or template. Each field provides information about an aspect of the requirement.
Design description languages	This approach uses a language like a programming language, but with more abstract features to specify the requirements by defining an operational model of the system. This approach is now rarely used although it can be useful for interface specifications.
Graphical notations	Graphical models, supplemented by text annotations, are used to define the functional requirements for the system; UML use case and sequence diagrams are commonly used.
Mathematical specifications	These notations are based on mathematical concepts such as finite-state machines or sets. Although these unambiguous specifications can reduce the ambiguity in a requirements document, most customers don't understand a formal specification. They cannot check that it represents what they want and are reluctant to accept it as a system contract

## Requirements and design

- به طور کلی، نیازمندی‌ها باید بیان کنند که سیستم چه کاری باید انجام دهد و طراحی باید توصیف کند که چگونه این کار انجام می‌شود.
- در عمل، نیازمندی‌ها و طراحی جدایی‌ناپذیر هستند؛
  - یک معماری سیستم ممکن است برای ساختاردهی نیازمندی‌ها طراحی شود؛
  - سیستم ممکن است با سایر سیستم‌ها که نیازمندی‌های طراحی را ایجاد می‌کنند، تعامل داشته باشد؛
  - استفاده از یک معماری خاص برای برآورده کردن نیازمندی‌های غیرعملکردی ممکن است یک نیازمندی دامنه باشد.
  - این ممکن است نتیجه یک نیازمندی قانونی باشد.

## Natural language specification

- نیازمندی‌ها به عنوان جملات به زبان طبیعی نوشته می‌شوند که با نمودارها و جداول مکمل شده‌اند.
- از زبان طبیعی برای نوشتن نیازمندی‌ها استفاده می‌شود زیرا بیانگر، شهودی و جهانی است. این بدین معناست که نیازمندی‌ها می‌توانند توسط کاربران و مشتریان درک شوند.

Dr. A. Taghinezhad

## Guidelines for writing requirements

- یک فرمت استاندارد ابداع کرده و آن را برای همه نیازمندی‌ها استفاده کنید.
- از زبان به روشی یکسان استفاده کنید. از "shall" برای نیازمندی‌های الزامی و "should" برای نیازمندی‌های مطلوب استفاده کنید.
- از هایلایت کردن متن برای شناسایی بخش‌های کلیدی نیازمندی استفاده کنید.
- از استفاده از اصطلاحات کامپیوتری اجتناب کنید.
- شامل یک توضیح (دلایل) برای ضرورت هر نیازمندی باشید.

## Problems with natural language

- عدم وضوح

□ دقت بدون دشوار کردن خواندن سند، دشوار است.

- اختلاط نیازمندی‌ها

□ نیازمندی‌های عملکردی و غیرعملکردی تمایل دارند

که با هم مخلوط شوند.

- ترکیب نیازمندی‌ها

□ چندین نیازمندی مختلف ممکن است با هم بیان

شوند.

## Example requirements for the insulin pump software system

**3.2** سیستم باید سطح قند خون را اندازه‌گیری کرده و در صورت نیاز، هر ۱۰ دقیقه انسولین تحویل دهد. (تغییرات در سطح قند خون نسبتاً کند هستند، بنابراین اندازه‌گیری‌های مکرر غیر ضروری است؛ اندازه‌گیری‌های کمتر ممکن است منجر به افزایش سطح قند خون به طور غیر ضروری شود.)

**3.6** سیستم باید هر دقیقه یک روتین خودآزمایی را اجرا کند که شرایطی که باید آزمایش شوند و اقدامات مرتبط در جدول ۱ تعریف شده‌اند. (یک روتین خودآزمایی می‌تواند مشکلات سخت‌افزاری و نرم‌افزاری را کشف کرده و کاربر را از اینکه عملکرد طبیعی ممکن است غیرممکن باشد، مطلع کند.)



## Structured specifications

- رویکردی به نوشتن نیازمندی‌ها که در آن آزادی نویسنده نیازمندی محدود است و نیازمندی‌ها به روشی استاندارد نوشته می‌شوند.
- این برای برخی انواع نیازمندی‌ها، مانند نیازمندی‌های سیستم کنترل تعبیه شده، خوب عمل می‌کند، اما گاهی برای نوشتن نیازمندی‌های سیستم تجاری بیش از حد سخت‌گیرانه است.

## Form-based specifications

- تعریف عملکرد یا موجودیت.
- توصیف ورودی‌ها و منبع آنها.
- توصیف خروجی‌ها و مقصد آنها.
- اطلاعات در مورد اطلاعات مورد نیاز برای محاسبات و سایر موجودیت‌های استفاده شده.
- توصیف عملی که باید انجام شود.
- شرایط پیش و پس (در صورت مناسب بودن).
- عوارض جانبی (در صورت وجود) عملکرد.

## Tabular specification

- برای مکمل کردن زبان طبیعی استفاده می شود.
- به ویژه زمانی که باید تعدادی از دوره های احتمالی جایگزین را تعریف کنید، مفید است.
- به عنوان مثال، سیستم پمپ انسولین محاسبات خود را بر اساس نرخ تغییر سطح قند خون انجام می دهد و مشخصات جدولی نحوه محاسبه نیاز به انسولین برای سناریوهای مختلف را توضیح می دهد.

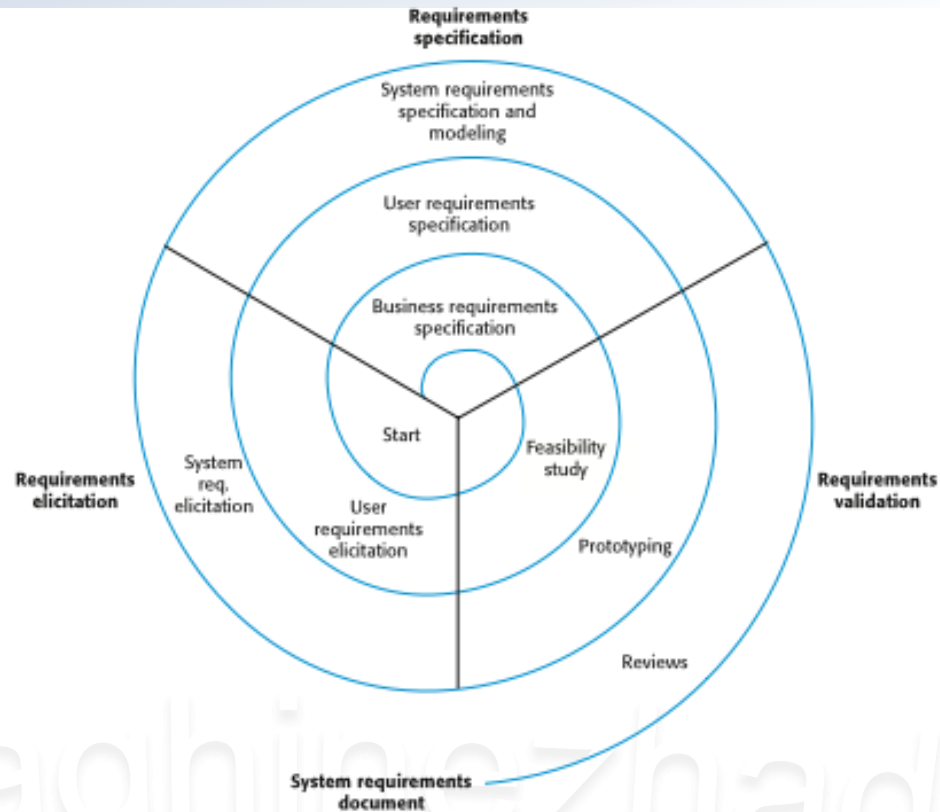
# Tabular specification of computation for an insulin pump

Condition	Action
Sugar level falling ( $r2 < r1$ )	CompDose = 0
Sugar level stable ( $r2 = r1$ )	CompDose = 0
Sugar level increasing and rate of increase decreasing ( $(r2 - r1) < (r1 - r0)$ )	CompDose = 0
Sugar level increasing and rate of increase stable or increasing ( $(r2 - r1) \geq (r1 - r0)$ )	CompDose = round $((r2 - r1)/4)$ If rounded result = 0 then CompDose = MinimumDose

## Requirements engineering processes

- فرآیندهای مورد استفاده برای مهندسی نیازمندی‌ها به شدت بستگی به دامنه کاربرد، افراد درگیر و سازمانی دارد که نیازمندی‌ها را توسعه می‌دهد.
- با این حال، چندین فعالیت عمومی در همه فرآیندها مشترک است؛
  - استخراج نیازمندی‌ها؛
  - تحلیل نیازمندی‌ها؛
  - اعتبارسنجی نیازمندی‌ها؛
  - مدیریت نیازمندی‌ها.
- در عمل، مهندسی نیازمندی‌ها یک فعالیت تکراری است که در آن این فرآیندها در هم تنیده می‌شوند.

# A spiral view of the requirements engineering process



Dr. A. Taghinezhad

## Requirements elicitation and analysis

- گاهی اوقات به آن استخراج نیازمندی‌ها یا کشف نیازمندی‌ها می‌گویند.
- شامل کارکنان فنی است که با مشتریان همکاری می‌کنند تا در مورد دامنه کاربرد، خدماتی که سیستم باید ارائه دهد و محدودیت‌های عملیاتی سیستم اطلاعات کسب کنند.
- ممکن است شامل کاربران نهایی، مدیران، مهندسانی که در نگهداری درگیر هستند، کارشناسان دامنه، اتحادیه‌های کارگری و غیره باشد. به این افراد ذینفعان گفته می‌شود.

## Problems of requirements analysis

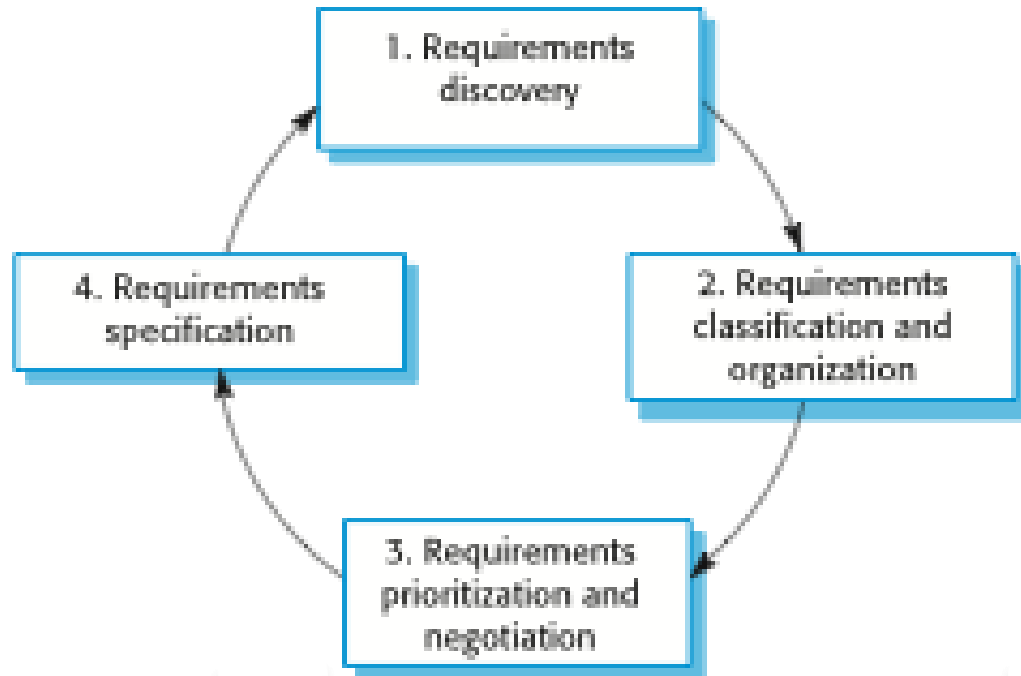
- ذینفعان نمی دانند که واقعاً چه می خواهند.
- ذینفعان نیازمندی ها را به زبان خود بیان می کنند.
- ذینفعان مختلف ممکن است نیازمندی های متعارضی داشته باشند.
- عوامل سازمانی و سیاسی ممکن است بر نیازمندی های سیستم تأثیر بگذارند.
- نیازمندی ها در طول فرآیند تحلیل تغییر می کنند. ذینفعان جدید ممکن است ظاهر شوند و محیط کسب و کار تغییر کند.



## Requirements elicitation and analysis

- مهندسان نرم‌افزار با طیف وسیعی از ذینفعان سیستم همکاری می‌کنند تا در مورد دامنه کاربرد، خدماتی که سیستم باید ارائه دهد، عملکرد مورد نیاز سیستم، محدودیت‌های سخت‌افزاری، سایر سیستم‌ها و غیره اطلاعات کسب کنند.
  - مراحل شامل:
    - کشف نیازمندی‌ها،
    - دسته‌بندی و سازمان‌دهی نیازمندی‌ها،
    - اولویت‌بندی و مذاکره نیازمندی‌ها،
    - مشخصات نیازمندی‌ها.

# The requirements elicitation and analysis process



Dr. A. Taghinezhad

## Process activities

- کشف نیازمندی‌ها
  - تعامل با ذینفعان برای کشف نیازمندی‌های آنها. نیازمندی‌های دامنه نیز در این مرحله کشف می‌شوند.
- دسته‌بندی و سازمان‌دهی نیازمندی‌ها
  - نیازمندی‌های مرتبط را گروه‌بندی کرده و آنها را به خوشه‌های منسجم سازمان‌دهی می‌کند.
- اولویت‌بندی و مذاکره
  - اولویت‌بندی نیازمندی‌ها و حل تعارض‌های نیازمندی‌ها.
- مشخصات نیازمندی‌ها
  - نیازمندی‌ها مستند شده و به دور بعدی از فرآیند چرخشی منتقل می‌شوند.

## Problems of requirements elicitation

- ذینفعان نمی‌دانند که واقعاً چه می‌خواهند.
- ذینفعان نیازمندی‌ها را به زبان خود بیان می‌کنند.
- ذینفعان مختلف ممکن است نیازمندی‌های متعارضی داشته باشند.
- عوامل سازمانی و سیاسی ممکن است بر نیازمندی‌های سیستم تأثیر بگذارند.
- نیازمندی‌ها در طول فرآیند تحلیل

## Key points

- سند نیازمندی‌های نرم‌افزاری یک بیانیه توافق شده از نیازمندی‌های سیستم است. این سند باید به گونه‌ای سازماندهی شود که هم مشتریان سیستم و هم توسعه‌دهندگان نرم‌افزار بتوانند از آن استفاده کنند.
- فرآیند مهندسی نیازمندی‌ها یک فرآیند تکراری است که شامل جمع‌آوری نیازمندی‌ها، مشخص‌سازی و اعتبارسنجی است.
- جمع‌آوری و تحلیل نیازمندی‌ها یک فرآیند تکراری است که می‌تواند به عنوان یک دور فعالیت‌ها نمایش داده شود - کشف نیازمندی‌ها، طبقه‌بندی و سازمان‌دهی نیازمندی‌ها، مذاکره درباره نیازمندی‌ها و مستندسازی نیازمندی‌ها.

NINTH EDITION

# Software Engineering

A PRACTITIONER'S APPROACH

ROGER S. PRESSMAN

BRUCE R. MAXIM

Mc  
Graw  
Hill

## Chapter 4 – Requirements Engineering

Part 3

## Requirements discovery

- فرایند جمع آوری اطلاعات و ذینفعان سیستم
- فرایند جمع آوری اطلاعات مربوط به سیستم‌های موجود و موردنیاز و استخراج الزامات کاربر و سیستم از این اطلاعات.
- تعامل با ذینفعان سیستم، از مدیران تا نهادهای نظارتی خارجی.
- سیستم‌ها معمولاً طیف گسترده‌ای از ذینفعان دارند.

Dr. A. Taghinezhad

## Stakeholders in the MHC-PMS

- بیمارانی که اطلاعاتشان در سیستم ثبت می شود.
- پزشکانی که مسئول ارزیابی و درمان بیماران هستند.
- پرستارانی که هماهنگی مشاوره ها با پزشکان را انجام داده و برخی از درمان ها را اجرا می کنند.
- مسئولان پذیرش پزشکی که قرارهای ملاقات بیماران را مدیریت می کنند.
- کارکنان فناوری اطلاعات که مسئول نصب و نگهداری سیستم هستند.
- مدیر اخلاق پزشکی که اطمینان حاصل می کند سیستم مطابق با دستورالعمل های اخلاقی مراقبت از بیماران است.
- مدیران بهداشت و درمان که اطلاعات مدیریتی را از سیستم دریافت می کنند.
- کارکنان ثبت سوابق پزشکی که مسئول حفظ اطلاعات سیستم و اجرای صحیح رویه های ثبت سوابق هستند.



# Interviewing

- مصاحبه‌ها در مهندسی نیازمندی‌ها
- مصاحبه‌های رسمی یا غیررسمی با ذینفعان بخشی از اکثر فرآیندهای مهندسی نیازمندی‌ها (RE) است.
- انواع مصاحبه‌ها:
  - مصاحبه‌های بسته بر اساس فهرستی از سؤالات از پیش تعیین شده.
  - مصاحبه‌های باز که در آن موضوعات مختلف با ذینفعان بررسی می‌شوند.
- مصاحبه مؤثر:
  - ذهنی باز داشته باشید، از پیش فرض‌ها در مورد نیازمندی‌ها اجتناب کنید و به ذینفعان گوش دهید.
  - با پرسیدن سؤال آغازین، ارائه پیشنهادات نیازمندی یا کار بر روی یک سیستم نمونه بحث‌ها را به جریان بیندازید.

## Interviews in practice

- ترکیبی از مصاحبه‌های بسته و باز معمولاً استفاده می‌شود.
- مصاحبه‌ها برای درک کلی از آنچه ذینفعان انجام می‌دهند و نحوه تعاملشان با سیستم مفید هستند.
- مصاحبه‌ها برای درک نیازمندی‌های حوزه خاص مؤثر نیستند:
  - مهندسين نیازمندی‌ها ممکن است اصطلاحات تخصصی را درک نکنند.
  - برخی دانش‌ها آنقدر آشنا هستند که افراد سخت می‌توانند آن‌ها را بیان کنند یا آن را مهم بدانند.

## Scenarios

- سناریوها نمونه‌های واقعی از نحوه استفاده از سیستم هستند. آنها باید شامل موارد زیر باشند:
  - توضیحی از وضعیت شروع؛
  - توضیحی از جریان عادی رویدادها؛
  - توضیحی از آنچه ممکن است اشتباه پیش برود؛
  - اطلاعاتی درباره سایر فعالیت‌های همزمان؛
- توصیفی از وضعیت زمانی که سناریو پایان می‌یابد

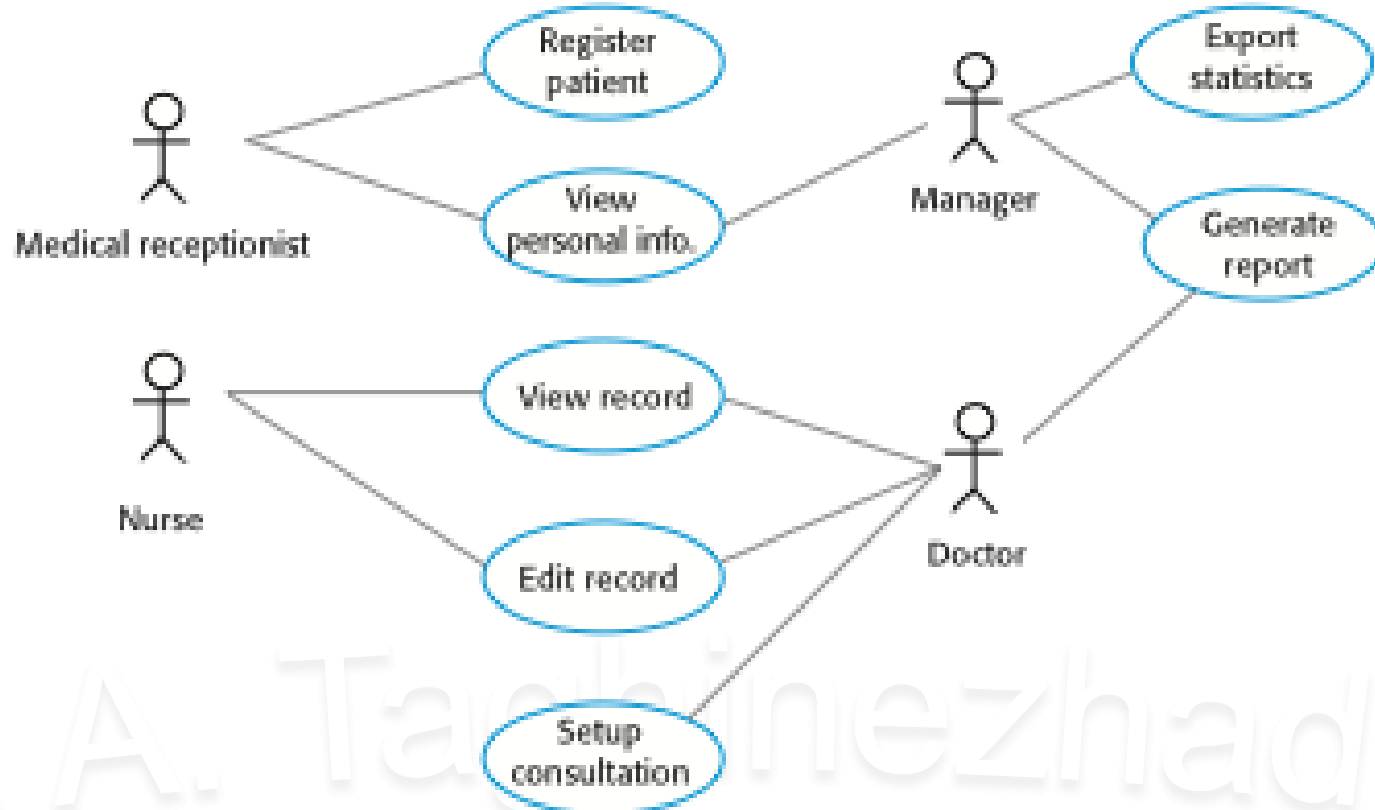
## Use cases

### • موارد کاربرد: (Use Cases)

- یک تکنیک مبتنی بر سناریو در UML که بازیگران در یک تعامل و خود تعامل را شناسایی می کند.
- یک مجموعه از موارد کاربرد باید تمام تعاملات ممکن با سیستم را توصیف کند.
- مدل گرافیکی سطح بالا که با توضیحات جدولی دقیق تر تکمیل می شود.
- برای افزودن جزئیات بیشتر، نمودارهای توالی می توانند ترتیب پردازش رویدادها را در سیستم نشان دهند.

Dr. A. Taghinezhad

# Use cases for the MHC-PMS



Dr. A. Taghinezhad

# Ethnography

- مطالعات قوم‌نگاری (Ethnography)
- یک دانشمند اجتماعی زمان قابل توجهی را صرف مشاهده و تحلیل نحوه کار واقعی افراد می‌کند.
  - افراد نیازی به توضیح یا بیان کار خود ندارند.
  - عوامل اجتماعی و سازمانی مهم ممکن است مشاهده شوند.
  - مطالعات قوم‌نگاری نشان داده‌اند که کار معمولاً غنی‌تر و پیچیده‌تر از آن چیزی است که مدل‌های ساده سیستم پیشنهاد می‌دهند.

# Ethnography

- تحلیل قوم‌نگاری در مهندسی نرم‌افزار
- تحلیل قوم‌نگاری (Ethnographic Analysis) یک روش تحقیق کیفی است که در تحلیل نیازمندی‌ها استفاده می‌شود. این روش شامل مشاهده کاربران در محیط واقعی آن‌ها برای درک بهتر رفتارها، فرآیندها و چالش‌هایشان است. هدف این است که نیازهایی را شناسایی کند که کاربران به‌طور مستقیم بیان نمی‌کنند.

- مثال کوتاه

- سناریو: طراحی یک سیستم مدیریت بیمار در بیمارستان.
- روش قوم‌نگاری: تحلیل‌گر به مدت چند روز پرستاران و پزشکان را مشاهده می‌کند که چگونه بیماران را ثبت می‌کنند و اطلاعات پزشکی را مدیریت می‌کنند.
- نتیجه: مشخص می‌شود که پرستاران برای پیدا کردن سوابق بیمار بین چندین سیستم مختلف سردرگم می‌شوند. این مشاهده به افزودن قابلیت "جستجوی یکپارچه" در سیستم منجر می‌شود.

## Scope of ethnography

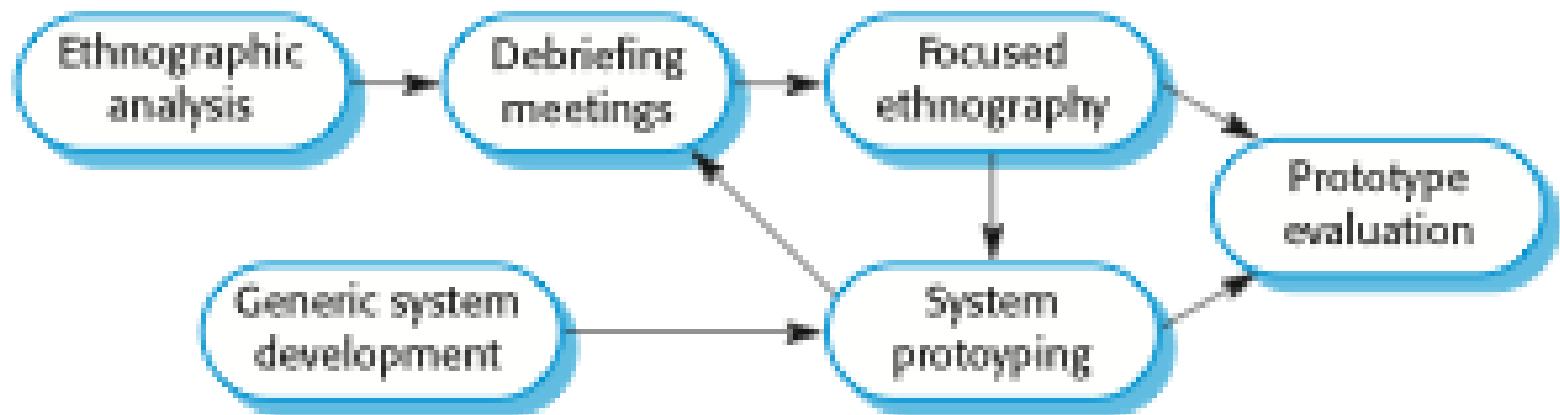
- الزامات مشتق شده از نحوه واقعی کار افراد به جای تعاریف فرآیند:
- الزامات ناشی از همکاری و آگاهی از فعالیت های دیگران.
- آگاهی از کار دیگران باعث تغییر در روش های کاری ما می شود.
- قوم نگاری برای درک فرآیندهای موجود مؤثر است اما نمی تواند ویژگی های جدیدی را که باید به سیستم اضافه شوند شناسایی کند.



## Focused ethnography

- ترکیب مردم‌نگاری با نمونه‌سازی
- توسعه‌شده در پروژه‌های برای مطالعه فرآیند کنترل ترافیک هوایی.
- ترکیبی از قوم‌نگاری (Ethnography) با نمونه‌سازی (Prototyping).
- توسعه نمونه اولیه منجر به سؤالات بی‌پاسخ می‌شود که تحلیل قوم‌نگاری را متمرکز می‌کند.
- مشکل قوم‌نگاری این است که رویه‌های موجود را مطالعه می‌کند که ممکن است مبنای تاریخی داشته باشند و دیگر مرتبط نباشند.

# Ethnography and prototyping for requirements analysis



Dr. A. Taghinezhad  
Dr. A. Taghinezhad

## Requirements validation

- بررسی صحت نیازمندی‌ها
- هدف: اطمینان از این که نیازمندی‌ها سیستمی را تعریف می‌کنند که واقعاً مورد نظر مشتری است.
- هزینه‌های خطا در نیازمندی‌ها بالا است، بنابراین اعتبارسنجی بسیار مهم است:
  - رفع خطاهای نیازمندی پس از تحویل ممکن است تا ۱۰۰ برابر رفع خطای پیاده‌سازی هزینه داشته باشد.

## Requirements checking

- **ارزیابی نیازمندی‌ها**
- **اعتبار**: آیا سیستم توابعی را فراهم می‌کند که نیازهای مشتری را به بهترین شکل پشتیبانی کند؟
- **ثبات**: آیا تعارضی بین نیازمندی‌ها وجود دارد؟
- **کامل بودن**: آیا تمام توابع موردنیاز مشتری شامل شده‌اند؟
- **واقع‌گرایی**: آیا نیازمندی‌ها با توجه به بودجه و فناوری موجود قابل پیاده‌سازی هستند؟
- **قابلیت اعتبارسنجی**: آیا نیازمندی‌ها قابل بررسی هستند؟

Dr. A. Taghinezhad

## Requirements validation techniques

- ابزارها و تکنیک‌های اعتبارسنجی نیازمندی‌ها
- بررسی نیازمندی‌ها: تحلیل دستی و سیستماتیک نیازمندی‌ها.
- نمونه‌سازی: استفاده از یک مدل اجرایی برای بررسی نیازمندی‌ها.
- ایجاد موارد تست: توسعه تست‌هایی برای بررسی قابلیت آزمون‌پذیری نیازمندی‌ها.

## Requirements reviews

- نکات مرتبط با بررسی نیازمندی‌ها
- بازبینی‌های منظم در طول تدوین تعریف نیازمندی‌ها برگزار شود.
- کارکنان مشتری و پیمانکار در بازبینی‌ها دخیل باشند.
- بازبینی‌ها ممکن است رسمی (با اسناد تکمیل شده) یا غیررسمی باشند. ارتباطات مؤثر بین توسعه‌دهندگان، مشتریان و کاربران می‌تواند مشکلات را در مراحل اولیه حل کند.

## Review checks

- خصوصیات نیازمندی‌ها
  - قابلیت آزمون‌پذیری: آیا نیازمندی واقعاً قابل آزمون است؟
  - قابلیت درک: آیا نیازمندی به درستی فهمیده شده است؟
  - ردیابی‌پذیری: آیا منشأ نیازمندی به وضوح بیان شده است؟
  - قابلیت انطباق: آیا نیازمندی بدون تأثیر عمده بر سایر نیازمندی‌ها قابل تغییر است؟

## Requirements management

- مدیریت نیازمندی‌ها
- فرآیند مدیریت تغییرات نیازمندی‌ها در طول مهندسی نیازمندی‌ها و توسعه سیستم.
- با پیشرفت سیستم و استفاده از آن، نیازمندی‌های جدید ظاهر می‌شوند.
- ضرورت پیگیری نیازمندی‌های فردی و حفظ ارتباط بین نیازمندی‌های وابسته برای ارزیابی تأثیر تغییرات.
- ایجاد فرآیند رسمی برای پیشنهاد تغییر و ارتباط آن با نیازمندی‌های سیستم.



## Changing requirements

### • تأثیر تغییرات محیطی و کاربران

• محیط تجاری و فنی سیستم همیشه پس از نصب تغییر می کند:

◦ سخت افزار جدید، تغییر اولویت های تجاری، یا قوانین جدید ممکن است سیستم را تحت تأثیر قرار دهند.

• افرادی که هزینه سیستم را پرداخت می کنند و کاربرانی که از آن استفاده می کنند معمولاً متفاوت هستند.

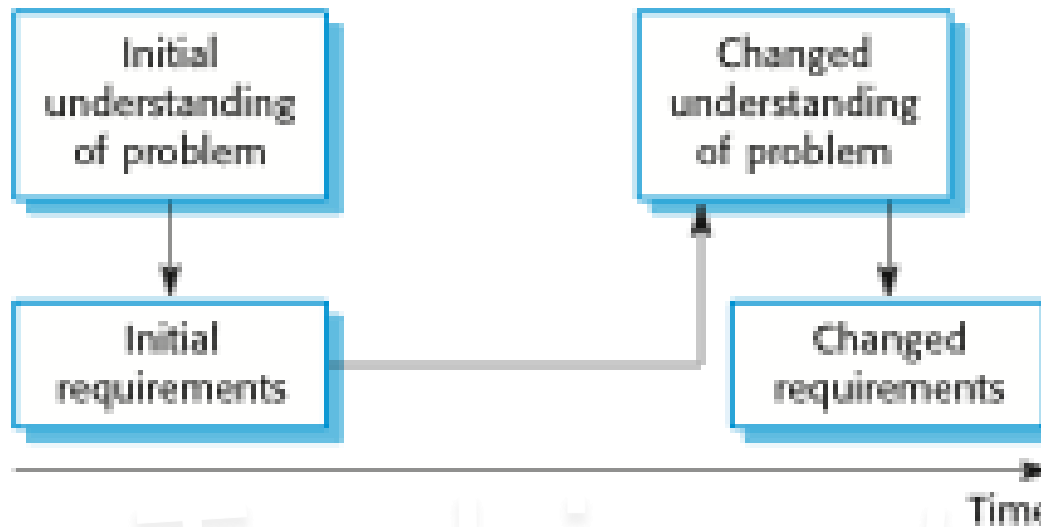
◦ نیازمندی های مشتریان ممکن است با نیازهای کاربران نهایی در تضاد باشد و پس از تحویل، ممکن است ویژگی های جدیدی برای پشتیبانی از کاربران اضافه شود.

## Changing requirements

- **تصمیم‌گیری برای پذیرش تغییرات نیازمندی‌ها**
- **تحلیل مشکل و تعیین تغییر:**
  - تحلیل پیشنهاد تغییر برای اطمینان از اعتبار آن.
- **تحلیل تغییر و هزینه‌گذاری:**
  - ارزیابی اثر تغییر پیشنهادی با استفاده از اطلاعات ردیابی و دانش عمومی سیستم.
- **اجرای تغییر:**
  - اصلاح مستندات نیازمندی‌ها و در صورت نیاز، طراحی و پیاده‌سازی سیستم.

Dr. A. Taghinezhad

# Requirements evolution



Dr. A. Taghinezhad

## Requirements management planning

- استفاده از تکنیک‌های مهندسی نیازمندی‌ها
- تکنیک‌های استخراج نیازمندی:  
شامل مصاحبه‌ها، سناریوها، موارد کاربرد (Use Cases) و قوم‌نگاری.
- اعتبارسنجی نیازمندی‌ها:  
بررسی اعتبار، سازگاری، کامل بودن، واقع‌گرایی و قابلیت اعتبارسنجی.
- مدیریت تغییرات:  
کنترل تغییرات ناشی از تغییرات تجاری، سازمانی و فنی.

## Requirements change management

- **تحلیل مشکل و مشخص سازی تغییرات:**  
در این مرحله، مشکل یا پیشنهاد تغییر تحلیل می شود تا مشخص شود آیا معتبر است یا خیر. نتایج این تحلیل به درخواست کننده تغییر بازگردانده می شود. ممکن است درخواست کننده، یک پیشنهاد تغییر مشخص تر ارائه دهد یا درخواست خود را پس بگیرد.
- **تحلیل تغییر و هزینه گذاری:**  
اثر تغییر پیشنهادی با استفاده از اطلاعات ردیابی و دانش کلی از نیازمندی های سیستم ارزیابی می شود. پس از تکمیل این تحلیل، تصمیم گرفته می شود که آیا تغییر نیازمندی اجرا شود یا خیر.
- **اجرای تغییر:**  
مستندات نیازمندی ها و در صورت نیاز، طراحی و پیاده سازی سیستم اصلاح می شود. ایده آل این است که مستندات به گونه ای سازمان دهی شوند که تغییرات به راحتی اجرا شوند.

# Requirements change management



Dr. A. Taghinezhad

## Key points

- می‌توانید از تکنیک‌های مختلف برای استخراج نیازمندی‌ها مانند مصاحبه‌ها، سناریوها، موارد کاربرد (Use Cases) و قوم‌نگاری استفاده کنید.
- اعتبارسنجی نیازمندی‌ها فرآیند بررسی اعتبار، سازگاری، کامل بودن، واقع‌گرایی و قابلیت اعتبارسنجی آن‌ها است.
- تغییرات تجاری، سازمانی و فنی به‌طور اجتناب‌ناپذیر به تغییر نیازمندی‌های سیستم نرم‌افزاری منجر می‌شود. مدیریت نیازمندی‌ها فرآیند مدیریت و کنترل این تغییرات است.