

بیشترین تأکید در رافن سندداشتن مشتری
↑
Communication

مروری بر جلسه چهارم

اصول تولید نرم افزار مبتنی بر فرآیند چابک

7. معیار اصلی پیشرفت پروژه، میزان پیشرفت فیزیکی نرم افزار تولید شده است.
8. در بازه های زمانی منظم و مشخصی تیم باید از لحاظ فنی خود را ارزیابی کرده و برای رسیدن به کارایی بیشتر، رفتار خود را بهبود دهد. (ارزیابی منظم تیم و محصول)
9. تیم های با انگیزه میتوانند نرم افزار با طراحی و معماری مناسبی تولید کنند.
10. باید بر سادگی تأکید شود.
11. تأکید مداوم به طراحی خوب و کار فنی خوب منجر به ارتقاء سرعت (Agility) خواهد شد.
12. فرآیند سریع، منجر به توسعه پایدار خواهد شد.

تیم درست همیشه را از برای کند تا بعد از آن با هم match شوند
تأکید بر سادگی است تا جای به سیمه SRS را سید کنیم
KISS: Kee it simple stupid
Solid design

اصول تولید نرم افزار مبتنی بر فرآیند چابک

1. اولویت اصلی راضی نگه داشتن مشتری و تولید نرم افزار با ارزش است.
2. در صورت ارائه درخواستهای جدید مشتری (درخواست تغییرات) حتی بعد از ارائه نسخه نهایی نرم افزار، از آن استقبال کرده و تغییرات مد نظر مشتری اعمال شود. (استقبال از تغییرات)
3. نرم افزار اولیه در طرف چند هفته تا چند ماه تولید شود. تأکید اصلی بر روی تولید و تحویل سریع نرم افزار (در بازه زمانی کوتاه) است.
4. هم مشتری و هم مهندسان نرم افزار باید روزانه باهم در رابطه با پروژه در ارتباط باشند. (ارتباط مشتری و تولید کننده)
5. افراد تیم باید انگیزه کافی برای کار کردن داشته باشند. به افراد تیم باید اعتماد کرد و نباید محیط کار مناسبی را برای اصلی آنها را برآورده کرد.
6. موثرترین روش برای ارتباط و انتقال اطلاعات در بین افراد تیم، روش بحث مستقیم و روزرو (چهره به چهره) است.

باید به تغییرات
توجه ویژه
تولید اولین نسخه سریع تر
داستور

احتمال اعتراض
تعمیم کمی تیم را سریع در صورت
بسرود، نامه نگاری کند
سرعتی بفرستیم برای تست و کارهای
ادارای پس از آن کار و نامه نگاری نکنند

در آفرانز مونس کار خود را بسوزرد
مشاوران خود را
هر روز بفرستیم برای هم بسوزرد

Sample Bank ATM	Version 1.0
Use-Case Specification: Withdraw Cash	Date: 01/07/2007

4. Basic Flow of Events

در درون کارت

4.1 Insert Card

The use case begins when the actor **Customer** inserts their **bank card** into the card reader on the ATM.
The system allocates an **ATM session identifier** to enable errors to be tracked and synchronized between the ATM and the Bank System.

4.2 Read Card

The system reads the **bank card information** from the card

خواندن اطلاعات کارت توسط سیستم

4.3 Authenticate Customer

Perform Subflow **Authenticate Customer** to authenticate the use of the **bank card** by the individual using the machine

بانک کاستور را شناسایی میکند

Subflow (خودش را شناسایی درود)

4.4 Select Withdrawal

The system displays the **service options** that are currently available on the machine.

مبلغ یا کاری که می خواهد

The Customer selects to withdraw cash.

انعام دهی یا سرویس آیسین مثلا در اینت پول نقد می

4.5 Select Amount

The system prompts for the amount to be withdrawn by displaying the list of **standard withdrawal amounts**.

اینجا مبلغ

The Customer selects an amount to be withdrawn.

4.6 Confirm Withdrawal

Perform Subflow **Assess Funds on Hand**

چک میکند کارت پول تو تن هست یا نه (کنترل)

Perform Subflow **Conduct Withdrawal**

پول را بهش برده

4.7 Eject Card

The system ejects the Customer's **bank card**.

The Customer takes the **bank card** from the machine

4.8 Dispense Cash

The system dispenses the requested amount of cash to the Customer.

پول را می دهد

کارت می برد

The system records a **transaction log** entry for the withdrawal.

4.9 Use Case Ends

The use case ends

مورد پس بعد از ۹ مرحله تمام کارود

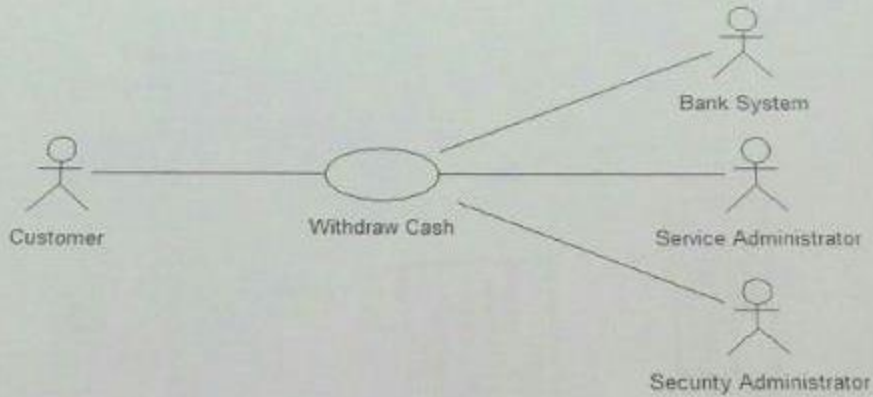
حالا مرحله ۱۰ مورد

Use-Case Specification: Withdraw Cash

1. Brief Description

This use case describes how a Bank Customer uses an ATM to withdraw money from a bank account.

2. Use-Case Diagram



3. Preconditions

- The bank Customer must possess a **bank card**.
- The network connection to the **Bank System** must be active.
- The system must have at least some cash that can be dispensed.
- The cash withdrawal **service option** must be available.

بین مشتری و سیستم

اول مشتری کارت باید داشته باشد

ارتباط عیار بین سیستم و کارت

در سیستم پول وجود داشته باشد

سرویس کارت کشی

4 : Base flow of Events (سازوکار اصلی رویداد)

مراحل : ورود کارت - خواندن کارت - شناسایی کارت - صدور اسکناس - اسکناس صادر

سازوکار اصلی
Sub Flow
استدلال شده



تایید Confirm

بعد از این عملیات استقانا صادر میگردد؟
 در صورتی که در سیستم پول وجود داشته باشد
 پس از این عملیات استقانا صادر میگردد؟

مروری بر جلسه دوم فرآیندهای تولید نرم افزار

- همه فرآیندهای ارائه شده فعالیت‌های مشترکی مانند موارد زیر را انجام میدهند
 - (Project Initiation - Requirement gathering) Communication
 - (Estimating - Scheduling - Tracking) Planning
 - (Analysis - Design) Modeling
 - (Code - Test) Construction
 - (Delivery - Support - Feedback) Deployment

■ اما هر کدام از این فرآیندها، فعالیت‌های مهندسی نرم افزار را به شیوه مختلف ارائه میکنند و یا اینکه ممکن است ترکیب و زمان اجرای هر کدام از این فعالیتها در فرآیندهای مختلف، متفاوت باشد

تکرار ماچا →
تیم پروژه پروژه - سازندگان
سکه چقدر نبرد - چقدر ساخت
های مدیریت
است
umbrella

آنجایی که ما در سیستم
بررسی می‌کنیم

مروری بر جلسه سوم فرآیندهای مهندسی نرم افزار

- مدل‌های کلاسیک (Traditional Software Models)
 - مدل آبشاری (Waterfall Model)
 - مدل V (V Model)
 - مدل افزایشی (Incremental Model)
 - مدل‌های تکاملی - مدل نمونه سازی (Prototyping Model)
 - مدل‌های تکاملی - مدل فزونی (Spiral Model)
 - مدل‌های تکاملی - مدل همزمان (Concurrent Model)
 - فرآیند یکپارچه (Unified Process)
 - روش‌های رسمی (Formal Methods)
 - مدل‌های مبتنی بر نمونه (Component Based Development)
- مدل‌های مدرن (Agile Software Modeling)

در همه مدل‌های حائز اعتبار مشترک

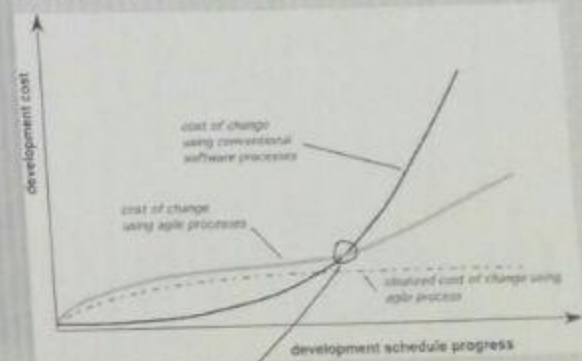
class diagram
Entity ها را شناس می‌کنند
شناختن DB ها را

① use-case - چه فعالیت‌هایی در سیستم داریم چه Activity می‌باشند

sequence - diagram
② جزئیات رفتار سیستم را مدل‌سازی می‌کنند
جزئیات روابط و داده‌سازی در use case

هزینه تغییرات که در شتاب با شرایط پروژه است

هزینه تغییرات در فرآیند سریع (Agile)



These slides are designed to accompany Software Engineering: A Practitioner's Approach, 7th (McGraw-Hill, 2008) slides copyright 2008 by Roger Pressman

سریعیت فوق تغییرات ، هزینه کمتر شده

مروری بر جلسه چهارم

چابکی (Agility) چیست؟

- پاسخ کارا و سریع به درخواستهای تغییرات در نرم افزار
- ارتباط موثر و کار با تمامی ذینفعان پروژه
 - ارتباط مداوم با مشتری
- درگیر کردن مشتری در تیم تولید نرم افزار
 - افزایش رضایت مشتری
- سازماندهی تیم به طوریکه بر کار مسلط باشند
 - تیم با انگیزه و کوچک

تولید سریع و افزایشی (تکراری) نرم افزار

مشارکت در هر مرحله سناریو را می نویسیم که باید در کمتر از چند هفته تکمیل شود و Increment

These slides are designed to accompany Software Engineering: A Practitioner's Approach, 7th (McGraw-Hill, 2008) slides copyright 2008 by Roger Pressman

در کمتر از چند ماه تکمیل شود

Increment باید تا این درگاه تکمیل داد شود

Agile تغییرات را هر چند سریعتر کند
چون هرگز معروف کند

در پس ها از مردمی که که تکمیل داد به هزینه بیشتر باشد
صبر باید که مشتری سود و از عوض بر آورد

از جایی که کمترین هزینه شده کامل نیست از آن به بعد دستیار کم تر می شود

اصول اساسی برای هدایت فعالیت‌های مهندسی نرم افزار (Practice Principles)

- اصل 4: ایجاد نرم افزار به صورت ماژول‌ها
- اصل 1 بر فلسفه استفاده شکستن مسائل به مسائل کوچکتر تاکید شد. برای تحقق بخشیدن به این اصل، باید نرم افزار را مبتنی بر قطعات (ماژول) ساخت.
- هر ماژول باید بر روی بخش مشخصی از نرم افزار تمرکز کند.
- اصل 5: استفاده از الگوها (Patterns)
- تکرار یک راه حل برای یک مشکل مشخص در مهندسی نرم افزار است.
- استفاده از الگو منجر به انتقال دانش و افزایش دانش در پروژه های مهندسی نرم افزار میشود.
- اصل 6: مسائل نرم افزار را از دیدگاههای مختلفی بررسی کنید
- مثلا نرم افزار میتوان بر اساس نوع داده، نوع کارکرد و رفتارهای آن مورد تحلیل قرار داد.
- اصل 7: به یاد داشته باشید که نرم افزاری که میسازید حتما قرار است توسط افرادی دیگر نگهداری شود.
- به دلیل تغییرات محیط و به وجود آمدن نیازمندیهای جدید ممکن است نیاز باشد تا قابلیت‌های چندینی به نرم افزار فعلی شما اضافه شوند لذا نرم افزار باید طوری ساخته شود که نگهداری آن آسان باشد.

اصول اساسی برای هدایت فعالیت‌های مهندسی نرم افزار

- هدف مهندسی نرم افزار، تولید به موقع نرم افزار با کیفیت به مشتری است. برای رسیدن به این هدف باید یک سری اصول، ارزشها و قوانین را در نظر گرفت. این اصول و قوانین مستقل از روشهای مختلف تحلیل، طراحی و کدنویسی خواهند بود.
- اصل 1: شکستن مسئله به مسئله کوچکتر (Separation of Concerns)
- در تحلیل و طراحی باید تاکید بر جداسازی و تقسیم مسائل و مشکلات به بخشهای جداگانه باشد.
- مسائل بزرگ در صورت شکسته شدن به مسائل ساده تر، بهتر مدیریت و حل خواهد شد.
- اصل 2: فهم استفاده از تجرید (abstraction)
- هدف استفاده از کلمات ساده برای مفاهیم پیچیده است.
- مثلا با شنیدن spreadsheet، شما به ساختار و قابلیت‌ها و کارکردهای آن پی خواهید برد.
- حذف جزئیات غیر ضروری در ارتباطات.
- اصل 3: تاکید و تمرکز بر انتقال دانش
- در هر نرم افزاری به نوعی انتقال دانش انجام می شود. انتقال دانش از پایگاه داده به کاربر، از کاربر به سیستم و ... در همه این موارد از طریق یک سری واسطه‌های اطلاعات رد و بدل می شود. لذا توجه ویژه به واسطه‌های مربوط به طراحی تحلیل و تست وجود دارد.

solid design
Keep it simple

سخت‌ترین کار در تقسیم کنیم

سخت‌ترین کار در تقسیم کنیم

مثلا همه word کلمات را

متوجه نمیشد چرا

کدام کلمات است

باید کلمات به معنای اولی

بسیار مهم

که از هم جدا هم کار کنیم، از شرکت جدا کنیم
افزود در این بخش کلمات و دانش ما کتد

structural & creative design

Sal Injection آیا برپایه سوزن؟

- تفاوت در سیستم، سرعت هم است مهم، سوزن را می‌توان تراشید، load جدید؟
- از side کان‌تیرا تراشید کار کتد و یا مدیریت در این کتد، نامش تغییر داد و کتد ... مهم است.

باید کلمات قابل دستار باشد

مستر دوست دارد همیشه نیازمند دروسان بودی که داده کلیدی سرد
نیازمند استاسب با بودی که مستری دارد
فصل ۵ کتاب

اصول دیگر

- در موضوعات دیگری نیز در مهندسی نرم افزار یک سری اصول مشخص وجود دارد. همه این اصول در فصل چهارم کتاب تشریح شده است. به تدریج در هر جلسه اصول باقیمانده که شامل موارد زیر است مورد بررسی قرار خواهد گرفت
- اصول برنامه ریزی (Planning Principles)
- اصول مدل سازی (Modeling Principles)
- اصول مدل سازی و تحلیل (Requirement Modeling Principles)
- اصول مدل سازی طراحی (Design Modeling Principles)
- اصول ساخت (Construction Principles)

مهندسی نیاز مندیها ۳۴۳۳۳

- Inception** (درک مسئله)
 - در این فاز که شامل کار است پرسش هایی مطرح می شود که موارد زیر را روشن می سازد:
 - ترسک کنی و باید از نرم افزاری که قرار است ساخته شود.
 - شناسایی افرادی که قرار است این نرم افزار برای آنها ساخته شود | شناسایی ذینفعان
 - مشخصات کلی مطلوب و مورد انتظار
 - شناسایی و آشنایی اولیه بین مهندسی نرم افزار و مشتری
- Elicitation** (اکتساب نیاز مندی ها)
 - از هر چه اساس اطلاع است بگیریم. برای آشناتراره نوشته شود
 - استاد این نرم افزار مدرسه
- Elaboration** (محل سازی نیاز مندی ها)
 - در این فاز نیاز مندی های اعلام ذینفعان استخراج می شود
 - در این فاز تجزیه و تحلیل صورت می گیرد نیاز مندی های سیستم حاصل سازی می شود. در این فاز داده ها توجیه
- Negotiation** (مذاکره)
 - در این فاز توسعه کننده و مشتری بر روی یک سیستم قابل ارائه و واقع ساده توافق می نمایند

می خواهیم برآورد کنیم. اول از همه باید ببینیم چه فرآیندی نوشته کنیم (چه مهندسی نیاز مندی ها) چه تا طبیعتی حاصل می باشد داشته باشد. هر چه از مستند سازی بود این نرم افزار را، به چه صورت

تغییر می دهد که لازم داریم داده ها را با هم ترکیب کنیم. این بین بر وجه قبول داشته باشد و حکم داشته باشد و ...

حالا تمسک می کنیم به روشی. شکل سیستم نمی می صورت. راه حل برای شکل سیستم، فرآیند یک نرم افزار جدید. این شکی نیست که باید برود. صورتی نیاز مندی ها به صورت ترکیبی داریم که سیستم مورد نیاز ما می که در هر چه سیستم نوشتیم. یک راه حل بدیم این راه حل خودش یک نرم افزار جدید با مشخصات جدید می شود. که آن نیاز مندی ها را حالا فعلی می کنیم با جزئیات کامل Sequence و یا نرم افزار جزئیات use-case ها را می کشیم

System = VRS
 activity collaboration sequence
 post condition
 هندسی نیاز مندیها

مهندسی نیاز مندیها

این اسلایدها طراحی شده برای توضیح به دست می آید. مستند نیاز مندی های نرم افزار (SRS) در این فرآیند مشخص می گردد و به توافق طرفین مشتری و تولید کننده می رسد. این مستند باید حداقل شامل یکی از موارد زیر باشد:

- لیست نیاز مندی ها به همراه اولویت آنها
- مجموعه ای از مدل ها
- مدل فرآیند مستند (فرآیند سیستم های آمیخته شده تبارکست)
- مجموعه ای از سناریوهای فرآیندهای سیستم (Use-Case)
- یک نمونه اولیه یا Prototype و نرم افزار

Validation (اعتبارسنجی)

شکل آن ضروری است

- فهمی های که در دسترس نیستند و نیاز به توضیح بیشتری دارند
- اطلاعات گمشده (Missing Information)
- ناسازگاری (یک مشکل عمده که در هنگام مهندسی سیستم ها و محصولات بزرگ پیش می آید)
- نیاز مندی های متناقض و یا بیرواقعی که برای هدف آنها باید با مشتری مذاکره شود.

همچنین بررسی فرآیند با استفاده از کنترل محتوای موارد

These slides are designed to accompany Software Engineering: A Practitioner's Approach, 7th Edition (McGraw-Hill, 2006). Slides copyright 2006 by Roger Pressman. 25

مهندسی نیاز مندیها

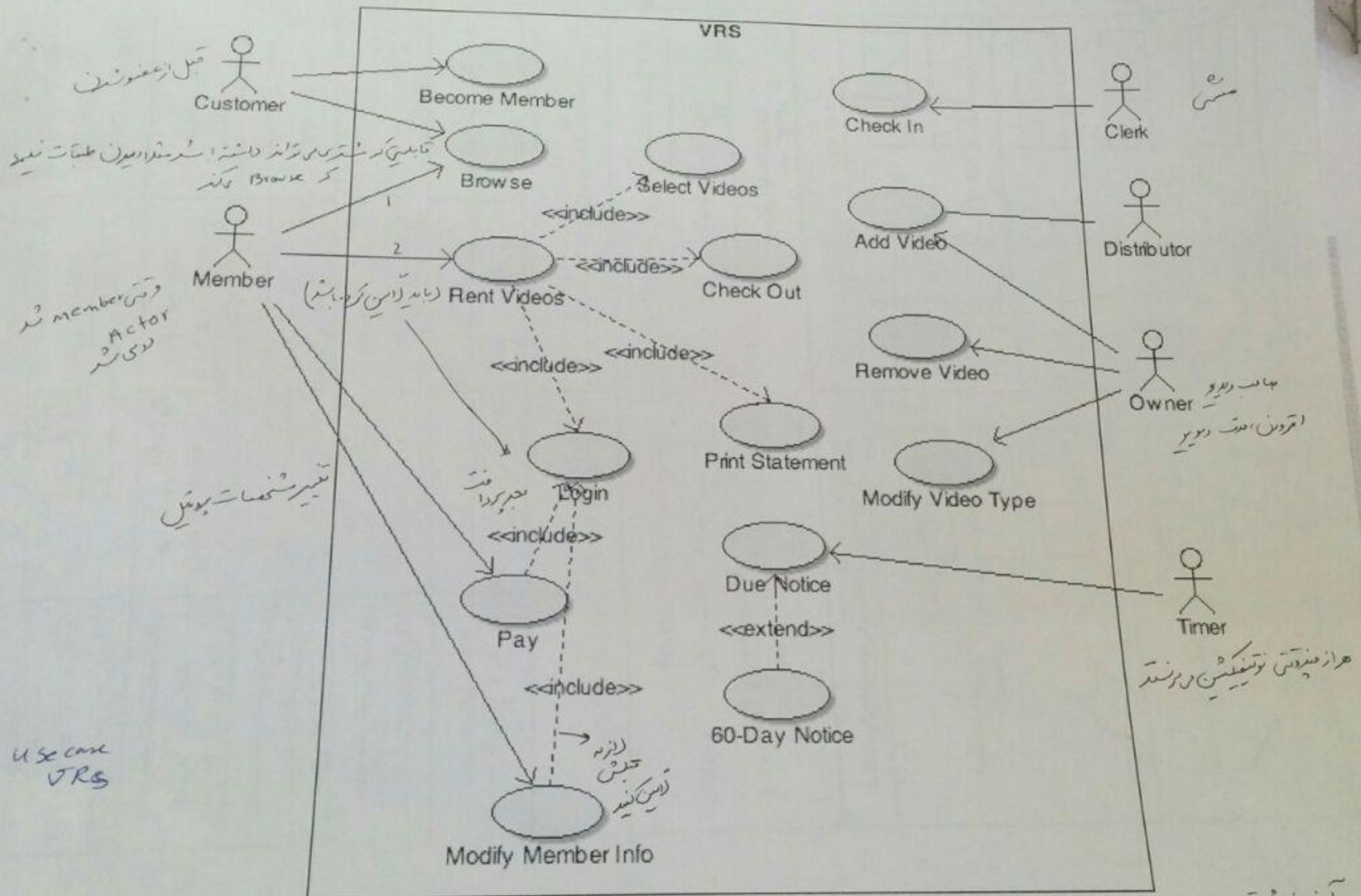
Requirements Management (مهندسی نیاز مندی ها)

فرآیند مهندسی نیاز مندی ها انجام می شود

These slides are designed to accompany Software Engineering: A Practitioner's Approach, 7th Edition (McGraw-Hill, 2006). Slides copyright 2006 by Roger Pressman. 26

Validation
 Verification
 Verification: کنترل محتوای موارد
 Validation: اطمینان از اینکه شکل درست است
 Verify: اطمینان از اینکه فرآیند درست است
 Validate: اطمینان از اینکه سیستم درست است
 Verify: اطمینان از اینکه فرآیند درست است
 Validate: اطمینان از اینکه سیستم درست است

specification: توضیح از use case
 use case: توضیح از use case
 collaboration sequence: توضیح از use case
 activity: توضیح از use case
 Specification: توضیح از use case



Use case VRS

آنگونه نوشتیم به زبان فارسی است و این هم Use case

5. Alternative Flows

5.1 Customer Authentication

5.1.1 Handle No Communications with the Bank System

At the **Validate Card Information** step of Subflow *Authenticate Customer* if the Bank System cannot be contacted or does not reply within the set **communication time out period**.

1. If the communications link has failed more times than the **communication retry number**, then the authentication attempt is abandoned and Basic Flow is resumed at **Use Case Ends**.
2. The system will attempt to contact the Bank System until it has completed the number of retry attempts indicated by the **communication retry number**.
3. If communications is re-established the Basic Flow is resumed at **Authenticate Customer**.
4. If there is still no response from the Bank System the system creates an **event log** entry to record the failure of the communications link to the Bank System. The **event log** entry includes the type of failure.
5. The system sends the **event log** to the **Service Administrator** to inform them that communications with Bank System has been lost.
6. Resume the Basic Flow at **Use Case Ends**.

5.1.2 Handle No Communications with the Customer's Bank

At the **Validate Card Information** step of Subflow *Authenticate Customer* if the Bank System reports that the **Customer's Bank** cannot be contacted,

1. The system creates an **event log** entry to record the fact that the **Customer's Bank** was unavailable. The **event log** entry includes the **bank card information** (excluding the PIN).
2. The system informs the Customer that communications with their Bank is not possible and that the Customer should try again later.
3. Resume the Basic Flow at **Use Case Ends**.

5.1.3 Handle Inactive Card or Account

At the **Validate Card Information** step of Subflow *Authenticate Customer* if the **Customer's Bank** reports that the card, or its associated **account**, are inactive.

1. The system creates an **event log** entry to record the fact that the Customer's account was inactive. The **event log** entry includes the **bank card information** (excluding the PIN).
2. The system informs the Customer that the account associated with the card is not active and that the Customer should contact their Bank for more information.
3. Resume the Basic Flow at **Use Case Ends**.

5.1.4 Handle Stolen Bank Card

At the **Validate Card Information** step of Subflow *Authenticate Customer* if the Bank System reports that the card has been stolen.

1. The system
 - a. Confiscates the card.
 - b. Captures a 10-second video image of the Customer.
 - c. Creates an **event log** entry to record the fact that a stolen card has been used. The **event log** entry includes the video image and the current **bank card information** (excluding the PIN).
 - d. Sends the **event log** entry to the Security Administrator, the Bank System and the Service Administrator to inform them that a stolen card is being used.

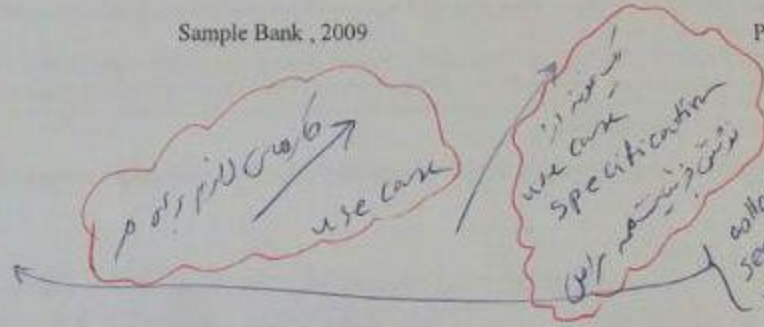
Handwritten notes in the top left margin.

Handwritten notes in the left margin, next to section 5.1.2.

Handwritten notes in the left margin, next to section 5.1.3.

Handwritten notes in the right margin, including a vertical list of items.

Handwritten note: "use case specification" with an arrow pointing to the left.



Handwritten note: "collaboration sequence activity" with an arrow pointing to the right.

Handwritten note at the bottom right: "use case" with an arrow.