مهندسی نرم‌افزار۲- جلسه ۱۱

نکات کلاسی

* ‌ برای ایجاد extension feature یا اضافه کردن قابلیت‌هایی در یک وب اپلیکیشن مانند بازی، چه زیر ساخت مناسبی در مهندسی نرم افزار وجود دارد؟
  + پاسخ:‌ باید یک زیرساختی داشته باشیم که extension توش راحت باشد - که برنامه نویسی براش آسان باشد. - باید یه زیرساختی داشته باشیم که بتونیم بهش وصل کنیم!.
  + زیر ساخت مورد نظر باید چه مشخصاتی داشته باشد؟
  + MIDE
* ویژگی MIDE ؟‌(‌ مطمئن نیستم)‌ ویژگی های ابزار‌ّای که تحت عنوان MIDE مطرح می‌شوند
  + باید general productivity feature داشته باشیم .
  + third party sdk integration
    - sdk : software development kit
    - یک سری third party هستند که یک سری feature های سیستم‌های مورد نظرمان را تولید میکنند و ما بهش می چسبونیم.
  + tools:
  + support:
    - امکان آشنایی با سیستم در مجموعه باشد
  + tutorial documantation
  + graphical user interface builder
  + مثال از MIDE: jango -
* mobile app midleware:
  + موجب ارتباط بین کامپوننت های مختلف میشوند. اصولا middle ware هستند برای هماهنگ سازی‌های کامپوننت های مختلف
  + تفاوت middle ware با interface:
    - middle ware اگر هماهنگ نباشد آنها را نیز هماهنگ میکنند
    - interface میشناسد
    - در interface دو طرف قضیه پیداست ولی در middle ware خیر
  + در moblie app تاکید داریم:
    - معماری : پیشنهاد می شود بر اساس سرویس باشد.
* سرویس: userinterface برای من نباشه - از یه مثلا برداریم. پس همان (‌کامپوننت)‌ کار را میکنیم ولی با سرویس و باید جایی باشد که قابل دسترسی باشد. (‌مثلا برای اپلیکیشن موبایل از اینا استفاده کنیم - توصیه میشود)
  + مشکل:‌اگر یکی از این سرویس‌ها مشکل پیدا کند -> سیستم خراب میشود ( وقتی هر سرویس چند عمل انجام میدهد)‌ -> راه حل هر سرویس یک عمل انجام میدهد
* کامپوننت:‌re-usable باشد. -> هزینه تولید بالاست . هم کامپونت ورودی داریم هم خروجی
* subprogram:
* subsystem:
* module: فانکشن سیستم را در ساختار \*\*\* برنامه در سخت افزار انجام دهد.
* میکروسرویس:‌ وقتی هر سرویس یک عمل انجام میدهد -> میکرو سرویس - بهترین دولوپمنت : میکرو سرویس داشته باشیم در جایی که سرویس‌ها انباشته شده باشند.
* تمرین **:** حداقل۲وبسایت۴۰۴رامطالعهنمایید
* تمرین۱۸**.**۶صفحه۴۰۹راحلکنید **(‌**بخش۱۸**.**۴کتاب**)**
* برای **interface design** حداقل۲ **reference** معرفینمایید**.**
* بحثبعدی **:‌ QUALITY :**
* **User experience :‌**
  + ما باید تجربیات یوزر را در رابطه با مسئله داشته باشیم
  + user experience تا ۷۹ در صد تا موفقیت وب اپلیکیشن موفق است
  + user exp نشان داده کسانی که در رابطه با یک اپلیکیشن موفق نباشند جواب سوالاتشون رو بگیرند، ۸۰ درصد آنها آن اپلیکیشن را کنار میگذارند
  + در طراحی نرم افزار موبایل توه به یوزر اکسپرینس :
    - تایید و تصدیق ( confirmation and acknowledgement) :‌ ما undo داریم که اگر اکی نبود undo کند
    - نباید حالت تهی داشته باشد:‌ کنترل مناسب برا سوالاتی که جواب ندارد.
    - در اپلیکیشن شما user exp نشان داده که باید پیمایش به روش کارا باشد. ( قابل اندازه گیری باشد)
* نکته ) در مهندسی نرم افزار history بسیار مهم است.
  + - دیگر باید توجه به context باشد :‌
    - تهیهاخواهندهشود **(‌**توسطخودتان**)**
    - با متن یا سوال ناقص جواب درست بدهد.
    - یکجزوهچندصفحهایدررابطهبا **user interface** برایموبایلکهدرآن **user exp** است**.** اینحتماخواندهشود

فصلکیفیتوکنترلکیفیتدرمهندسینرم**‌**افزار **(‌**فصل۱۹**) -** خیلیجاهاازکتابمیریمبیرون**!** ولیکتابپوششدادهمیشود**.**

* کیفیت چیست؟ ابعاد کیفیت ؟‌هزینه کیفیت؟
* برای اینکه هرچیزی رو تشخیص بدیم :‌ scope + not scope :‌ سیستم
* پس کیفیت +‌عدم کیفیت = تعریف کیفیت
* مثلا، یکی از عدم کیفیت :‌ از دست دادن داده‌ها - functionality سیستم دقیق و صیخیخ و reliable نباشد- ذیع النفعان سیستم نیازاشون برطرف نشه - defectهای سیستم برطرف نشه:دی
* سیستم با کیفیت اولا باید کار کند. ( سیستم که کار نکند اصلا سیستم نیست)
* سیستم موفق: ؟
* سیستم با کیفیت است که با هزینه تعریف شده - زمان تعریف شده - کیفیت تعریف شده ارائه شود.
* کیفیت با مدل کیفیت تعریف میشود.
* شاخص وقتی شاخص است که بتوان اندازه گیری شود.
* نکته )‌ هر چی خرج میکنیم باید بازده داشته باشد ( اصل صفر)
* برای ساختن مدل کیفیت و … باید موارد زیر را بلد باشیم‌:
  + پیشبینی زمان
  + پیشبینی هزینه
  + اندازه گیری
  + مانیتورینگ:) (‌ مطمئن نیستم)
    - * تمرین**:‌** فریمورکسیستمصحیحبافریمورکسیستمباکیفیتچهتفاوت**‌**هاییدارد؟
* فریم ورک چیست؟ یکسری اکتیویتی است. در فریم ورک یک سیستم با کیفیت یکسری اکتیویتی اضافی داریم که همه‌ی آنها از جنس umbrella می باشد.
  + مشکل : یک سیستم با کیفیت که همه چیز آن خوب است ممکن است از نقطه نظر‌های مختلف متفاوت به نظر بیاید. یعنی یکی خوشش بیاد . یکی نه! (‌دید کاربر و ….)
* ابعاد کیفیت از نظر گاروین:
  + performance
  + feature qualityآیا نرم افزار خروجی ‌های قابل توجهی برای user دارد؟
  + reliability
  + conformane: - SEI - IS—IEEE- CMM
  + do ability:
* ابعاد کیفیت از نظر آقای mccall:
  + بعد product revision یا revision: یعنی متریکس ها - (‌ از کتاب بخون - نهفمیدم)- توی اسلاید ها هم هست قاعدتا