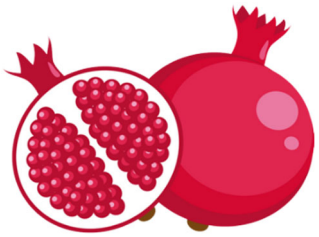


دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

12/14/2020

تمرین چهارم

مهندسی نرم افزار ۲

گروه {۲}

اعضاء گروه:

۱) محمدرضا اخگری زیری - ۹۶۳۱۰۰۱

۲) محمدعلی کشاورز - ۹۶۳۱۰۶۱

۳) علی نظری - ۹۶۳۱۰۷۵

۱) درباره متریک های نرم افزار[1] مطالعه و تحقیق کنید.

الف) متریک نرم افزار چیست؟

پاسخ: متریک نرم افزار، بررسی و اندازه گیری ویژگی‌های قابل اندازه‌گیری یا قابل شمارش نرم‌افزار است.
به کمک این متریک‌ها میتوان عملکرد نرم‌افزار را بررسی کرد؛ برای برنامه ریزی کارهای آینده برنامه‌ریزی کرد؛ بهره‌وری را اندازه‌گیری کرد و... به همه این علت‌های ذکر شده، متریک‌های نرم‌افزار مهم هستند.
متریک‌های نرم‌افزار به صورت در هم تنیده هستند؛ یعنی همه آن‌ها به نحوی به دیگر متریک‌ها مربوط هستند.
متریک‌های نرم افزار همگی بر اساس ۴ ویژگی مدیریتی برنامه ریزی، سازماندهی، کنترل و بهبود طراحی می‌شوند.
از مزایای بررسی و تجزیه و تحلیل متریک‌های نرم‌افزار، تعیین کیفیت محصول تولید شده (یا در صورت تولید نشدن محصول، کیفیت وضعیت فرآیند فعلی) و بهبود کیفیت و پیش‌بینی کیفیت محصول نهایی است. هدف نهایی از این بررسی‌ها بهبود بازگشت سرمایه، شناخت مناطق بهبود، تنظیم فشار و حجم کار، کم کردن کارهای اضافی و کاهش هزینه‌ها است.
نکته مهمی که در بررسی و مانیتور کردن متریک‌های نرم‌افزار وجود دارد این است که اگرچه این متریک‌ها در تعیین اهداف و اندازه‌گیری عملکرد و... اثر مثبت دارند، اما ساده‌سازی بیش از حد آنها باعث می‌شود که توسعه‌دهندگان از هدف اصلی خود که ارائه نرم‌افزار مفید و مورد رضایت مشتری است دور شوند. مثلا اگر معیار بهره‌وری را تعداد خط و کم بودن خطا بگذاریم ممکن است که توسعه‌دهندگان اقدام به توسعه خطوط کد زیاد اما ساده کنند که موجب می‌شود از هدف اصلی آن ها را دور کند. بنابراین اگرچه بایستی معیارها را تا حدی ساده کرد، اما از ساده‌سازی بیش از حد بایستی جلوگیری کرد.

ب) یک دسته بندی برای انواع متریک ها پیشنهاد کرده و برای هر نوع یک مثال بزنید.

پاسخ: برای دسته بندی متریک‌های نرم‌افزار، می‌توان دسته‌بندی زیر را ارائه کرد:

- متریک های برآورد هزینه و تلاش
  - روزهای فعال: میزان زمان مشارکت دولوپر در توسعه نرم‌افزار است (این زمان شامل زمان پلنینگ و... نیست)
- متریک‌های بررسی بهره‌وری
  - متریک‌های مرتبط با اندازه: این متریک‌ها بر اساس اندازه کد و با واحد کیلو خط کد (هزار خط کد) که با نماد KLOC بیان می‌شود، نمایش داده می‌شود. سه تا از مهم‌ترین متریک‌های مرتبط با اندازه میانگین تعداد خطا در هر KLOC، میانگین تعداد نقص در هر KLOC و میانگین هزینه برای هر KLOC است.

[1] Software Metrics

- متریک های قابلیت اطمینان
  - میزان خرابی برنامه(ACR): به حاصل تقسیم تعداد کرش‌کردن برنامه بر تعداد استفاده از برنامه، میزان خرابی برنامه می‌گویند.
- متریک‌های بررسی ساختار و پیچیدگی
  - متریک‌های عملکردگرا(Function-oriented metrics): بر اساس میزان عملکردهایی که توسط نرم‌افزار ارائه می‌گردد، تمرکز می‌کند، اما از آن‌جایی که عملکرد به صورت مستقیم قابل اندازه‌گیری نیست از موضوعی به نام نقطه عملکرد (function point) استفاده می‌شود (واحد اندازه گیری که عملکرد تجاری ارائه شده توسط محصول را کمی می‌کند را نقطه عملکرد می‌گویند)
- متریک‌های ارزیابی بلوغ نرم‌افزار
  - نرخ باز و بسته‌شدن ایشوها(Open/close rates): به تعداد ایشوهای مرتبط با ارور که در بازه زمانی ثابت ارائه می‌شود.

منبع:

https://stackify.com/track-software-metrics/#:~:text=A%20software%20metric%20is%20a,productivity%2C%20and%20many%20other%20uses

۲) Twelve-Factor App[2]یک متدولوژی برای ساخت برنامه های ‘نرم افزار به عنوان خدمت[3]’ است، در مورد این متدولوژی مطالعه کرده و به سوالات زیر پاسخ دهید.

الف) هر کدام از فاکتورهای دوازده گانه این متدولوژی کدام فاکتورهای کیفیتی ISO/IEC 9126[4] را پشتیبانی می کند؟ چگونه؟ دلایل خود را به طور کامل شرح دهید.

پاسخ:

- Codebase
  - Functionality بله زیرا به نظرم تحت عنوان interoperability، ممکن است این ریپوی ما با سایر ابزارها همگام شود و کاری برای ما انجام دهد. مثلا با یک سرویس که خطاهای ما را نشان می‌دهد همگام شود و بگوید که تغییر در کدام کامیت منجر به این خطا شده است.
  - Portability بله زیرا با وجود یک ریپو برای یک سرویس، مسائل مربوط به انتقال و جایگزینی، راحت‌تر انجام می‌شوند.

[2] https://12factor.net
[3] Software as a Service (SaaS)
[4] کتاب Pressman۱۹.۲.۳ (نسخه ۸):

- Dependencies
  - Functionality بله زیرا با مشخص بودن نسخه‌های پکیج‌های نرم‌افزاری استفاده‌شده از خطاهای احتمالی در راه‌اندازی مجدد سرویس جلوگیری می‌شود.
  - Usability بله زیرا به نظرم ممکن است به خاطر مشخص نبودن نسخه‌ها و پکیج‌های استفاده شده، بعضا دچار اشتباهاتی شویم که قابل فهم بودن سرویس زیر سوال برود.
  - Maintainability بله زیرا در ممکن است پایداری سرویس به خاطر استفاده از نسخه‌های اشتباه زیر سوال برود.
  - Portability بله زیرا با مشخص بودن پکیج‌ها می‌توان به راحتی سرویس را در جای دیگر نیز استفاده کرد.

- Config
  - Functionality بله زیرا مثلا هاردکد کردن کانفیگ‌ها درون کد ممکن است حتی امنیت را خدشه‌دار کند.
  - Usability بله زیرا به نظرم با جدا بودن کانفیگ‌ها می‌توان نسخه‌های مختلفی از آن برای کارهای متفاوت درست کرد و operability را افزایش می‌دهد.
  - Efficiency بله زیرا راحت‌تر می‌توان مسائل مربوط به دیپلوی را مدیریت کرد و از این لحاظ بهینه‌تر است.
  - Maintainability بله زیرا تغییر کانفیگ‌هایی که به صورت مستقل درآمده اند بسیار ساده‌تر است.
  - Portability بله، با توجه به موارد قبلی این مورد نیز کاور می‌شود چون هم سادگی به همراه داشته هم امکان تغییرات راحت‌تر را برای ما به ارمغان آورده است.

- Backing services
  - Functionality بله زیرا همکاری بین سرویس‌های جانبی راحت‌تر و بهتر انجام می‌شود.
  - Reliability بله زیرا به نظرم می‌توان به صورت مستقل پایداری آن‌ها را کنترل کرد و در نتیجه پایداری کلی سیستم بالاتر می‌رود.
  - Usability بله زیرا می‌توان راحت‌تر ارتباط‌ها را فهمید و آن‌ها را مدیریت کرد.
  - Efficiency بله به نظرم هم در time behavior و هم در resource behavior موثر است.
  - Maintainability بله زیرا می‌توان یک سرویس جانبی را به راحتی تغییر داد.
  - Portability بله، دلیلم همانند مورد قبلی است.

- Build, release, run
  - Reliability بله زیرا به راحتی می‌توان نسخه‌ای که قبلا بیلد شده را در صورت مشکل جایگزین کرد.
  - Usability بله زیرا به نظرم فرایندی که رخ می‌دهد این‌گونه شفاف‌تر و قابل فهم‌تر است.
  - Efficiency بله زیرا به نظرم از نظر ریسورسی مثلا ما یک بار یک چیز را بیلد می‌کنیم و می‌توانیم بعدا و هرچند بار که خواستیم از آن استفاده کنیم.
  - Maintainability بله زیرا مثلا می‌توانیم چیزی را که یک بار بیلد کردیم در محیط استیجینگ تست کنیم و اگر مشکلی نداشت همان را به عنوان نسخه اصلی ریلیز و ران کنیم.
  - Portability بله به نظرم زیرا باز هم همان بیلد قابلیت استفاده در هر جایی را دارد.

- Processes
  - Functionality بله زیرا مثلا از مشکلات inconsistency که ممکن است استیت‌ها ایجاد کنند جلوگیری می‌شود.
  - Reliability بله زیرا به نظرم کلا کمتر شدن استیت‌ها از خطا جلوگیری می‌کند و در نتیجه پایداری افزایش می‌یابد.
  - Efficiency بله زیرا به نظرم الکی حافظه‌های اضافی استفاده نمی‌شود.
  - Maintainability بله زیرا باز هم کم‌بودن استیت‌ها می‌تواند باعث پایداری بیشتر شود.

- Port binding
  - Functionality بله زیرا به نظرم کار رو توی همکاری بین سرویس‌ها ساده‌تر می‌کنه.
  - Maintainability بله زیرا به راحتی می‌شه با عوض کردن یک پورت کار رو به سرانجام رسوند.
  - Portability بله زیرا تغییر رو ساده‌تر می‌کنه.

- Concurrency
  - Functionality بله زیرا به نظرم با اسکیل‌کردن و بهبود مثلا تاخیرها، دقت کار بالاتر می‌رود.
  - Reliability بله زیرا پایداری سرویس وقتی که اسکیل‌پذیر باشن خیلی بهتر میشه.
  - Maintainability بله دیگه به وضوح پایداری بیشتر میشه.

- Disposability
  - Reliability بله زیرا وقتی یه مشکلی پیش بیاد و سرویس بیاد پایین، میشه سریع بالا آوردش و مشکل رو حل کرد.

- Efficiency بله زیرا قشنگ از لحاظ زمانی به ما کمک می‌کنه.
- Maintainability بله زیرا میشه پایدارتر باقی ماند.
- Portability بله زیرا می‌توان در مدت زمان کمی در صورت نیاز جا به جایی انجام داد.

- Dev/prod parity
  - Functionality بله زیرا در شرایط مشابه، کار و عمکردها یکسان اند و دقت کار بالا می‌رود.
  - Reliability بله زیرا ممکن است با تغییرات باعث ایجاد ناپایداری شویم.
  - Usability بله زیرا در صورت وجود تغییر، فهمیدن مشکلات هم سخت‌تر می‌شود.
  - Maintainability بله زیرا آنالیز مواردی که دچار تغییر شده‌اند بسیار سخت‌تر است.

- Logs
  - Usability بله زیرا با دیدن لاگ‌ها می‌توان همه چیز را بهتر فهمید.
  - Maintainability بله زیرا با دیدن لاگ‌ها می‌توانیم به صورت پیوسته مسائل را بررسی کنیم.

- Admin processes
  - Functionality بله زیرا به نظرم اگر مواردی که کارهای مدیریتی و مربوط به ادمین است مستقل باشند باعث افزایش امنیت می‌شود.
  - Reliability بله زیرا به نظرم می‌توان در صورت بروز مشکل می‌توان به صورت مستقل و سریع‌تر مشکل را حل کرد.
  - Maintainability بله زیرا مانند مورد قبلی می‌تواند باعث افزایش پایداری ما شود.

ب) آیا این متدولوژی کیفیت برنامه های نرم افزار به عنوان خدمت را تضمین می کند؟ به طور کامل توضیح دهید.

پاسخ: این ۱۲ فاکتور موارد مهمی بودند که امروزه در شرکت‌های خوب تلاش می‌شود در اکثر نرم‌افزارها به آن‌ها توجه شود ولی کیفیت نرم‌افزارها لزوما با این موارد تضمین نمی‌شود.

استفاده از یک ریپازیتوری، مشخص کردن پکیج‌های استفاده شده، جدا کردن کانفیگ‌های مربوط به دیپلوی و... اکثر این موارد را حتی در پروژه‌های کوچک نیز می‌بینیم ولی لزوما کیفیت خوبی ندارند و مشکلات بزرگ دیگری دارند.

در نهایت به نظرم این موارد در راستای تضمین کیفیت نرم‌افزار ما هستند ولی به صورت کامل هم این کیفیت را کامل و تضمین نمی‌کند.

منبع:

https://12factor.net/

۳) برای هر یک از انواع تست زیر، یک ابزار پیشنهاد کنید: آن را معرفی کنید، امکانات و نحوه کارکرد آن را شرح دهید، همچنین توضیح دهید که ابزار نیازهای (خاص) نوع تست مربوطه را چگونه فراهم می کند.

الف) تست رابط کاربری

پاسخ: برای آزمون رابط کاربری، ابزار متعددی مثل Ranorex، Selenium، QTP، Cucumber، Silk Test، TestComplete، Squish GUI Tester و ... وجود دارد که ما Selenium را انتخاب کردیم.

یکی از بهترین ابزارهای تست عمل‌کرد وب‌سایت، Selenium است که با این نرم‌افزار، حین طراحی وب‌سایت، نه تنها می‌توانید تست‌هایی را برای هر بخش تعریف کنید، بلکه می‌توانید تست‌ها را زمان‌بندی کنید و به طور خودکار انجام دهید. فرض کنید وب‌سایت شما ۱۰۰ ویژگی مختلف دارد، شما همه‌ی این ۱۰۰ ویژگی را تست و وب‌سایتتان را منتشر می‌کنید. بعد از دو روز یک باگ گزارش می‌شود یا اینکه می‌خواهید در روند فعالیت کاربر تغییر کوچکی ایجاد کنید. آیا پس از اعمال تغییرات یا رفع باگ، یا به صورت کلی حین طراحی سایت خود، می‌خواهید دوباره همه‌ی آن ۱۰۰ ویژگی را تست کنید؟ اینجاست که سلنیوم به کمک شما می‌آید و همه‌ی تست‌ها را به صورت خودکار انجام می‌دهد.

سلنیوم یک چارچوب تست نرم‌افزاری قابل حمل برای برنامه‌های تحت وب است. امکان طراحی تست وب‌سایت را بدون نیاز به دانش زبان اسکریپت تست فراهم می‌کند. همچنین یک زبان خاص دامنه (Selenese) را برای نوشتن تست در تعدادی از زبان‌های برنامه‌نویسی وب محبوب مانند C#، Groovy، Java، Perl، PHP، Python، Ruby و Scala فراهم می‌کند. پس از آن، تست‌ها را می‌توان در بسیاری از مرورگرهای وب اجرا کرد.

سلنیوم دو بخش اصلی به نام‌های سلنیوم IDE یا (selenium IDE) و سلنیوم وب درایور (selenium webdriver) دارد. سلنیوم IDE یک افزونه برای مرورگر است و پس از نصب آن با کلیک بر روی آیکنش، سلنیوم شروع به ضبط فعالیت‌های شما می‌کند (همه کلیک‌ها، پرکردن فرم‌ها و...). حتی خودتان هم می توانید به آن دستوراتی بدهید، مثلا گرفتن اسکرین شات از صفحه! تمامی این دستورات در مستند این ابزار واقع در این سایت نوشته شده است (برای یادگیری نحوه استفاده از آن هم داکیومنتی در انتهای این بخش به زبان فارسی قرار داده شده است و هم آموزش به زبان فارسی در آپارات).

همانطورکه اشاره شد، استفاده راحت از این ابزار و همچنین تولید unit test در زبان‌های مختلف دلیل انتخاب ما برای این ابزار است.

برای آموزش این ابزار تست به زبان فارسی، لینک آپارات مناسب است.

لینک آموزش استفاده از این ابزار در اینجا قرار دارد.

ب) تست API

پاسخ: برای تست API، ابزارهای متعددی وجود دارد، از جمله:

- Katalon
- Soap UI
- PostMan
- Tricentis Tosca
- Apigee
- JMeter
- REST-assured
- Assertible
- Karate
- Swagger

هر کدام از این ابزار، فواید و معایبی دارند، ما ابزار Postman را انتخاب کردیم.

- آسان جهت استفاده از کلاینتREST
- رابط کاربری غنی که استفاده از آن را آسان می‌کند
- می‌تواند برای هر دو آزمایش خودکار و اکتشافی مورد استفاده قرار گیرد
- قابل اجرا بر روی برنامه‌های Mac، Windows، Linux و مرورگر chrome است
- دارای تعدادی از یکپارچگی مانند پشتیبانی از فرمت های Swagger و RAML است
- دارای ویژگی های اجرا، تست، اسناد و مانیتورینگ
- نیازمند به یادگیری یک زبان جدید نیست
- کاربران را قادر می‌سازد تا دانش را به راحتی با تیم به اشتراک بگذارند زیرا می‌توانند تمام درخواست‌ها و پاسخ‌های مورد انتظار را به همکاران خود ارسال کنند
- پشتیبانی از GraphQL پس از نسخه ۷.۲

طریقه تست کردن API در این لینک ویرگول نوشته شده است که می‌توانید مشاهده کنید.

ج) تست مقاومت در برابر اشکال[5] در محیط ابر

پاسخ: برای تست مقاومت در برابر اشکال ابزار متعددی وجود دارد، از جمله:

- **CHAOS MONKEY**
- **Varien**
- **Pantera**

که ما ابزار chaos monkey را معرفی می‌کنیم که Netflix از آن استفاده می‌کند. ولی قبل از آن به chaos engineering می‌پردازیم.

Chaos Engineering یک متودولوژی در صنعت توسعه‌ی نرم‌افزار است که با استفاده از آن دولوپرها این اطمینان را حاصل می‌کنند که اپلیکیشن پس از دیپلوی شدن روی سرورهای اصلی چقدر توانایی تحمل شرایط غیرطبیعی و نامعمول را دارا است که چنین قابلیتی اصطلاحاً تحت عنوان Resiliency شناخته می‌شود؛ به عبارتی، سرویس آنلاین ما باید به گونه‌ای طراحی شده باشد که به خوبی از پسِ مشکلات زیرساختی، شبکه‌ای و نرم‌افزاری برآید. به زبانِ ساده، Chaos Engineering به توسعه‌دهندگان امکانی می‌دهد تا آنچه را که فکر می‌کنند پس از ایجاد یک مشکل در سرورها رخ خواهد داد را به خوبی اعتبارسنجی کنند.

در سال ۲۰۱۰ نت‌فلیکس به سمت ابر مهاجرت کرد و این در حالی بود که در فضای ابری این نیاز احساس می‌شد تا بتوان سرورها را در هر زمانی کانفیگ، تعمیر و تعویض کرد و نیاز به توضیح نیست برای سرویسی همچون نت‌فلیکس که در هر لحظه میلیون‌ها کاربر آنلاین دارد چنین کاری چالش‌های خاص خود را داشت که در همین راستا مهندسین این شرکت در سال ۲۰۱۱ ابزار Chaos Monkey را طراحی کردند.

Chaos Monkey ابزاری اپن‌سورس است که توسط مهندسین کمپانی Netflix طراحی شده است که این امکان را در اختیار توسعه‌دهندگان قرار می‌دهد تا عمداً بخش‌هایی از سرویس آنلاین همچون ماشین‌های مجازی یا کانتینرهای خود که روی AWS قرار دارند را از کار بیندازند تا در نهایت متوجه شوند که اگر اپلیکیشن واقعاً در چنین شرایطی قرار گیرد چه‌قدر انعطاف‌پذیر است (این ابزار با سرویس‌هایی همچون AWS ، Google Compute Engine، Azure، Kubernetesو Cloud Foundry سازگار است).

چنانچه یک میمون وحشی را در یک دیتاسنتر رها کنیم، آنچنان خرابی‌های زیادی به بار خواهد آورد که می‌تواند یک سرویس آنلاین جهانی همچون نت‌فلیکس را از کار بیندازد و Chaos Monkey نیز از همین قرار است.

Chaos Monkey کاملاً قابل‌برنامه‌ریزی است بدین صورت که امکان شبیه‌سازی مشکل در سرورها را در بازه‌های زمانی مشخصی به دولوپرها می‌دهد به طوری که پس از ایجاد مشکل به صورت عمدی، مهندسین می‌توانند رفتار

[5] Fault Tolerance

سیستم را از نزدیک رصد نمایند تا در نهایت آمادگی لازم برای مواقعی که چنین مشکلاتی به صورت واقعی صورت پذیرند را بیابند.

پس از موفقیت‌آمیز بودن این ابزار، کمپانی نت‌فلیکس مجموعه ابزارهایی تحت عنوان Simian Army را به منظور عملیاتی کردن انواع و اقسام تست‌های نرم‌افزاری در فضای ابری را ابداع کرد که Chaos Monkey نیز یکی از آن‌ها است که مجموع این ابزارها به مهندسین این شرکت کمک می‌کنند تا امنیت، انعطاف‌پذیری و همچنین قابل‌اعتماد بودن اپلیکیشن را تست کنند (به طور مثال، Chaos Gorilla یکی از زیرشاخه‌های این مجموعه نرم‌افزارها است که مسئولیت شبیه‌سازی از دست خارج شدن دیتاسنتر در یک ناحیه‌ی جغرافیایی خاص را بر عهده دارد).

برای آموزش استفاده از این ابزار می‌توانید به صفحه مستند این ابزار در گیت‌هاب مراجعه کنید.

۴) تست استرس[6]، تست بارگذاری[7] و تست غوطه وری[8]، از انواع تست های کارایی[9] هستند.

الف) آن ها را معرفی کرده و کارکرد هر کدام را شرح دهید.

تست استرس یا تست فشار: به ما اجازه می‌دهد که وضعیت بار نهایی که در آن عملکرد سیستم غیرقابل‌قبول است را تعیین کنیم. در این نوع از تست، کارآیی سیستم تحت بارهای افزایشی تدریجی اندازه‌گیری می‌شود. این به تسترها کمک می‌کند که نقطه نهایی که سیستم در آن fail می‌شود را تعیین کنند.

بار سنگینی که می‌توان به برنامه وارد کرد می‌تواند شامل مقادیر زیر باشد:

- مقادیر عددی پیچیده
- مقادیر زیاد ورودی
- مقادیر زیاد پرس‌وجو

هدف از این تست، طراحی محیطی مخرب‌تر از محیطی که برنامه در دنیای واقعی و در شرایط نرمال با آن روبرو می‌شود، است.

تست بارگذاری: تست بار یک روش آزمایش عملکرد است که با استفاده از آن پاسخ سیستم در شرایط مختلف بار اندازه‌گیری می‌شود. آزمایش بار برای شرایط نرمال و اوج بار انجام می‌شود. تست بار برای درک این موضوع است که سیستم نرم‌افزاری ما تحت شرایط واقعی بار همچنان به خوبی کار می‌کند یا خیر.

تست غوطه‌وری: آزمایش سیستم برای مدت زمان طولانی با باری در حد انتظار یا کمی بیشتر از بار مورد انتظار است. ایده پشت تست این است که ممکن است در طی تست‌های کوتاه، سیستم سریع پاسخ دهد اما در طولانی مدت به علت برخی از مشکلات حافظه که پس از مدت زمان طولانی مشخص خواهد شد عملکرد سیستم دچار مشکل شود.

ب) آن ها را با یکدیگر مقایسه کنید.

در قسمت قبلی با توضیح تقریبا فرق این موارد مشخص شد، اما بیشتر اوقات تست بارگذاری و فشار شبیه به هم هستند که در ادامه آن‌ها را مقایسه می‌کنیم.

تست استرس یا تست فشار، برای تعیین کارایی سیستم در حالتی که بار سیستم در حالت بیشینه باشد، موردستفاده قرار می گیرد. در واقع برنامه در مقابل بار سنگینی مانند مقادیر عددی پیچیده، مقادیر زیاد ورودی و مقادیر زیاد پرس و جو امتحان می‌شود، تا با مواجه ساختن برنامه با موقعیت‌های غیرمعمول و تزریق بار سنگین، میزان تحمل

[6] Stress Test
[7] Load Test
[8] Soak Test
[9] Performance

برنامه بررسی و تست شود. ولی تست بار برای تعیین کارایی سیستم در حالتی که بار سیستم به صورت طبیعی باشد، مورد استفاده قرار می‌گیرد.

ج) دو ابزار تست کارایی را معرفی کرده و چهار مورد از امکانات هر کدام را بیان کنید.

Load Ninja:

مورخ ۲۹ اکتبر ۲۰۱۸ مصادف با ۷ آبان ماه ۱۳۹۷ شرکت SmartBear، به عنوان یک شرکت پیشرو در ابزارهای کیفیت نرم‌افزار برای تیم‌ها، LoadNinja را به عنوان یک پلتفرم Cloud-Base برای مهندسان و کارشناسان حرفه‌ای و متخصصان Performance که اپلیکیشن‌های وب را تست بارگذاری می‌نمایند، منتشر کرد. LoadNinja یک ترکیب نادر ازEfficiency، دقت و داده‌های Performance عملیاتی را ارائه می‌دهد. به این ترتیب تیم‌ها می‌توانند تست بار را به راحتی در یک محیط Agile و DevOps ادغام کنند. بر خلاف پلتفرم‌های Load Test سنتی که مبتنی بر پروتکل هستند، نیازمند برنامه‌نویسی خسته کننده و ارتباط جنبه‌های پویای اپلیکیشن می‌باشند.LoadNinja به مهندس مربوطه اجازه می‌دهد که بیشتر به ساخت اپلیکیشن بپردازد و کمتر روی ساخت Load Testing Scriptها متمرکز شود.

بر خلاف بارِ تولید شده توسط ابزارهای معمولی که از شبیه‌سازهای مروگر اختصاصی و توسعه یافته توسط فروشنده استفاده می‌کنند، تکنولوژی TrueLoad در LoadNinja از مرورگرهای واقعی در مقیاس مناسب برای تست‌های بار استفاده می‌کند و واقع‌بینانه‌ترین و دقیقترین بار روی زیرساخت پشتیبانیِ برنامه وب تحت تست را ایجاد می‌کند.

علاوه بر موارد مذکور، LoadNinja دو نوآوری دیگر را در قالب VU Inspector و VU Debugger فراهم کرده است VU Inspector .به شما اجازه می‌دهد تا Degradation (تنزل رتبه) را به تصویر بکشید، چرا که کاربران مجازی در برابر برنامه بار تولید می‌کنند. هنگامی که یک کاربر مجازی با یک مشکل مواجه می‌شود، VU Debugger می‌تواند برای اتصال و ارتباط با مرورگر کاربر مجازی که دچار مشکل شده است به کار آید، تا بدین ترتیب برای جمع آوری اطلاعات تشخیصیِ تکمیلی در مورد مشکل استفاده شود.

به طور کلی چهار مورد از امکاناتش به شرح زیر است:

- برای تست load احتیاجی به نوشتن script نیست
- اجرای آزمایش بار توسط مرورگرهای واقعی
- بر پایه ابر است
- استفاده از VU debugger و VU inspector

JMETER

نرم‌افزار Apache JMeter یک نرم‌افزار Open Source و یک برنامه جاوایی ۱۰۰٪ خالص (Pure Java Appication) است که برای اِعمال تست بار روی رفتارهای Functional و سنجش میزانPerformance، طراحی شده است. این ابزار در اصل برای تستWeb Application ها طراحی شده است اما به دیگرTest Functionها نیز گسترش یافته است.

این ابزار قادر است یک بار سنگین روی یک سرور، گروه سرورها، شبکه یا Object را شبیه‌سازی نماید تا دوام و استحکام بخش تحت تست را بررسی کرده و یا Performance کلی را زیر انواعLoad های مختلف تحلیل کند.

امکانات Apache JMeter عبارتند از:

- توانایی برای تست Load و Performance بسیاری از انواع مختلف اپلیکیشن‌ها/سرورها/پروتوکل‌ها
- قابل حمل و پشتیبانی از تمام برنامه‌های مبتنی بر جاوا
- نمودارهای ساده برای تجزیه و تحلیل آمار مربوط به بار اصلی و مانیتورهای استفاده از منابع
- پشتیبانی از مجموعه جمع کننده Tomcat در زمان واقعی برای مانیتورینگ

منابع:

Top 10 Performance Testing Tools | Load Testing Tools Guide | Edureka
API.ir نیست (Browser)یک مرورگر JMeter- چیست؟ - وب-سرویس JMeter
(tisten.ir) بازآفرینی کرد – تیستن LoadNinja ها را باWeb Application برای (Load Testing)تست بارگذاری SmartBear

۵) درباره تست وقفه[10] مطالعه کنید. (امتیازی)

الف) آن را مختصرا معرفی کنید.

پاسخ: تست وقفه شاخه ای از تست برنامه‌های موبایل است که با نحوه واکنش یک برنامه در برابر وقفه و از سرگیری و بازگشت به حالت قبلی خود سروکار دارد.

ب) ده سناریوی وقفه برای برنامه های موبایل بیان کنید.

پاسخ:

- رفتن به اپلیکیشن دیگری در هنگام کار با آن اپلیکیشن (به اصطلاح بردن اپلیکشن به بک‌گراند مود)
- دریافت تماس

[10] Interrupt Testing

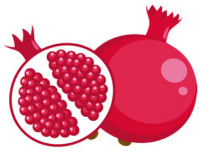
- دریافت پیامک یا پیام واتس‌اپ
- رفتن به حالت اسلیپ
- قفل خودکار سیستم
- رخ‌دادن آلارم
- هشدار کم‌شدن شارژ
- به شارژ زدن گوشی
- ایجاد نوتیفیکیشن مربوط به آپدیت نرم‌افزار
- هشدار پرشدن حافظه جانبی
- زدن به شارژ
- کشیدن از شارژ

ج) دو ابزار برای این نوع تست معرفی کنید.

پاسخ:

UIAutoMonkey: این ابزار برای ایجاد تست‌های استرس و تست وقفه در تلفن همراه طراحی شده است که با مراجعه به ریپازیتوری آن می‌توان جزئیات بیشتری از این ابزار را دید.

Mautomate: ابزار دیگری که برای تست وقفه استفاده می‌شود Mautomate است که توانایی ریکورد کردن تست‌های موبایل را دارد. برای آشنایی بیشتر با این ابزار نیز می‌توان به این لینک مراجعه کرد.



- پاسخ تمرین ها را به زبان فارسی و به صورت تایپ شده، در قالب یک فایلPdf ، در مودل بارگزاری کنید.
- سوالات خود را می‌توانید از طریق ایمیل از دستیاران تدریس بپرسید.
- فایل پاسخ تمرین را تنها با قالب **SE2-HW4-GroupNumber.pdf** در مودل بارگزاری کنید.
- بارگزاری تمرین توسط یکی از اعضاء گروه کافی است.
- برای پاسخ‌های هر قسمت منابع استفاده شده را درج نمائید.
- فایل زیپ ارسال **نکنید**.
- به ازای هر روز تاخیر در تحویل تمرین ۲۰٪ از نمره تمرین کسر خواهد شد.
- حداقل برخورد به پاسخ‌های مشابه، تخصیص نمره کامل منفی به طرفین خواهد بود.