گزارش عملکرد شماره ۱

تعاریف و مفاهیم اولیه

معماری نرم‌افزار:

تعریف رسمی معماری نرم‌افزار تحت استاندارد IEEE ۱۴۷۱-۲۰۰۰ به صورت مقابل است: "معماری نرم افزار یک قسمت اصلی از سیستم است، که شامل مولفه های سیستم، ارتباطات آن ها یا یکدیگر، محیط و اصول حاکم بر طراحی و ارزیابی آن می باشد."

معماری نرم‌افزار در چرخه ۶ مرحله‌ای عمر نرم‌افزار در مرحله سوم یعنی طراحی، به صورت طراحی معماری قرار دارد. در این چرخه مراحل به ترتیب مهندسی سیستم، تحلیل نیازهای نرم‌افزاری، طراحی، پیاده‌سازی، آزمون و نگهداری می‌باشند. هدف ما در این پژوهش انجام بخشی از مرحله آزمون پس از مرحله طراحی سیستم می‌باشد تا بتوان قبل از پیاده‌سازی بسیاری ار مشکلات را رفع نمود.

باید توجه داشت که بین معماری‌ها مختلف در زمینه‌های ابعاد، فرآیند، هزینه، زمان‌بندی، مهارت‌ها و تیم‌های توسعه، تکنیک‌ها، سهام-داران، ریسک‌ها تفاوت وجود دارد که این تفاوت‌ها می‌توانند در تشخیص کلاس‌های آزمون مبتنی بر مدل نهایی ما تاثیر مستقیم داشته باشد.

آزمون و آزمون مبتنی بر مدل:

به طور کلی هدف آزمون تشخیص خطا است. یعنی مشاهده یک تفاوت مابین رفتار آنچه که پیاده‌سازی شده است با آنچه که بر مبنای مشخصات انتظار داریم. آزمون در گذشته به صورت سنتی انجام می‌پذیرفت و معمولا آزمون کننده‌ها با ایجاد اسکریپت های ایستا، کار آزمون را انجام می دهند که بعد از یک زمان مشخص کل سیستم پوشش داده می‌شود. مشکل این حالت در زمان تغییر رفتار نرم‌افزار می‌باشد که در آن حالت تمامی کارهای آزمون باید از ابتدا صورت می‌پذیرفت. راه حل آن استفاده از آزمون مبتنی بر مدل می‌باشد.

امروزه مدل‌سازی، نقش حیاتی در فرآیند ایجاد نرم‌افزار بر عهده دارد. یک مدل یک توصیف از رفتار سیستم است. مدل‌ها به صورت ساده‌تر یک سیستم را شرح می‌دهند؛ همچنین به ما کمک می‌کنند تا رفتار یک سیستم را تشخیص دهیم و پیش‌بینی کنیم. از جمله مدل‌های مورد نیاز برای آزمون مبتنی بر مدل UML، FSM، درخت تصمیم و... می‌باشد.

آزمون مبتنی بر مدل به آزمون نرم‌افزاری اشاره دارد که نمونه‌های آزمون از تمام یا بخشی از مدل تشریح کننده سیستم تحت آزمون (SUT) مشتق شده‌اند. ایده اصلی آزمون نرم‌افزارهای مبتنی بر مدل، مشخص کردن وساخت مدل(های) خلاصه برای نمایش خواص مشخص و رفتار SUT است.

مهمترین دلایل استفاده از آزمون مبتنی بر مدل را می‌توان موارد زیر برشمرد:

- امروزه، آزمون نرم‌افزار در پروژه‌های نرم‌افزاری ۳۰ تا ۵۰ درصد کل بودجه پروژه نرم‌افزار را شامل می‌شود.
- با افزایش پیچیدگی نرم‌افزارها، آزمون آن‌ها نیز پرهزینه‌تر و زمان‌بر می‌شود.
- توانایی خودکارسازی عمل آزمون، در آزمون مبتنی بر مدل مهمترین ویژگی حرکت به سمت این نوع آزمون است.

بعضی مزایای آزمون مبتنی بر مدل عبارتند از:

- نگهداری راحت از نمونه ی آزمون
- کاهش هزینه همراه با آزمون بیشتر
- توانایی انجام آزمون های مختلف برروی هزاران ماشین
- تشخیص سریع باگ ها
- افزایش تعداد باگ ها
- حفظ زمان
- رضایت مندی بیشتر فرد آزمون کننده
- انجام عمل آزمون به صورت خودکار از همان ورژن ابتدایی ۰.۱
- فهم بهتر از SUT با ساخت و آنالیز مدل
- توان اعمال آزمون رگرسیون
- انجام آزمون MBT به صورت خودکار
- کاهش خطاهای انسانی در نیازمندی های سیستم و طراحی

بعضی از معایب آزمون مبتنی بر مدل هم عبارتند از:

- مهمترین مشکل این روش اجبار برای رسمی بودن ساختار و مدل مرجع می باشد.
- مجموعه مهارت های اضافه
- مدل ها ممکن است یک سرمایه گذاری درست و معنی دار باشند که هرگز تمامی باگ ها را پیدا نمی کنند.
- معیار سنجش آن مناسب نیست اما قابل تغییر است:
  معیار سنجش بد: تعداد باگ ها، تعداد نمونه های آزمون
  معیار سنجش بهتر: پوشش مشخصات(spec coverage)، پوشش کد(code coverage)

فرمال‌سازی:

در علم کامپیوتر، مخصوصا مهندسی نرم افزار، روش های رسمی(صوری) تکنیک مبتنی بر ریاضیات برای مشخصه سازی توسعه و اعتبارسنجی نرم افزار و سیستم های سخت افزاری می باشد.

بررسی مزایا و معایب روشهای استفاده شده در کارهای مشابه:

معایب	مزایا	عنوان کار
میزان خلاصه کردن مدل مرجع نسبت به پیاده‌سازی خیلی زیاد	برای سیستم‌های concurrent، real-	Using Software Architecture for Code Testing [۱]

است/ نوع روشADL و LTS و زبان برنامه‌نویسی گفته نشده است/ ابزار استفاده شده تنها برای سیستم‌های ایستا است./ خودکار نبودن تولید نمونه‌های آزمون / سیستم به طور کامل مورد آزمون قرار نمی‌گیرد.	specification-based ،time testing قابل استفاده است./ استفاده هم‌زمان از دو روش رسمی/ استفاده از ALTS	
نمونه‌های آزمون خودکار به دست نمی‌آیند./ برای این روش پیاده-سازی وجود ندارد.	**استفاده از مدل رسمی معماری به عنوان مدل مرجع برای آزمون مبتنی بر مدل/ امکان خودکارسازی تولید نمونه‌های آزمون/ استفاده از آتاماتای تیمی برای رسمی سازی معماری نرم‌افزار**	A New Model-Based Test Technique Using Formal Specification of Software Architectures [۷]
ACME بیشتر برای معماری‌های ایستا می‌باشد.	**امکان تولید خودکار نمونه های آزمون/ ترکیب روش‌های Petri Net و زبان ADL برای رسمی‌سازی**	Model Based Testing Using Software Architecture [۵]
در سطح کلی رها شده و کاربرد آن مشخص نیست./ تنها برای ضوابط آزمون بیان شده است و در آن مجبور به محدودسازی بازه‌ی کاری هستیم./ زبان رسمی‌سازی آن زبان Z است که در عین سادگی برای سیستم‌های پیچیده مشکل‌ساز است و با مشکل انفجار مولفه‌ها رو به رو می‌شود/ زبان Z به اندازه ی سایر زبان‌ها مانند Alloyقابلیت آنالیز ندارد.	**تمامی شرایط و ضوابط آزمون نرم-افزار به صورت رسمی بیان می‌شود./ تعاریف اصلی رسمی سازی در آن بیان شده است.**	Formalization of Software Testing Criteria using the Z Notation [۸]

هرس کردن تعداد زیادی از اهداف آزمون برای راحت‌تر شدن پیاده‌سازی/ خلاصه بودن بیش‌از اندازه نمونه‌های آزمون و ارائه تنها یک نمونه آزمون به عنوان خروجی هر هدف آزمون	این روش پیاده‌سازی شده است./ امکان تولید خودکار نمونه‌های آزمون در آن موجود است.	Fastest: a Model-Based Testing Tool for the Z Notation[۹]
تنها معماری نرم‌افزارهای پویا مورد بررسی قرار گرفته است	استفاده از زبان توصیف معماری π-ADL/ پوشش معماری‌های پویا و همراه	Dynamic Software Architectures: Formally Modeling Structure and Behaviour with π-ADL[۱۰]
ضعیف بودن در زمینه معماری‌های ایستا	ارائه یک زبان توصیف معماری رسمی کارآمد به همراه کامپایلر مخصوص خود این زبان/ تحت پوشش قرار دادن تمامی دیدگاه‌ها معماری از جمله ساختاری و رفتاری	π-ADL: An Architecture Description Language based on the Higher-Order Typed π-Calculus for Specifying Dynamic and Mobile Software Architectures[٤]