

دانشکده مهندسی کامپیوتر و فناوری اطلاعات
دانشگاه صنعتی امیرکبیر

توسعه RUP جهت ایجاد نرم افزارهای مطمئن

میثم حجازی نیا 86131117

ارائه پروژه درس طراحی سیستمهای مطمئن

برنامه آموزش مجازی- مقطع کارشناسی ارشد- گرایش نرم افزار

استاد درس: دکتر دهقان

نیمسال دوم 1386

هدف ارائه

- نرم افزار مطمئن

- سیستم های توزیع شده

- RUP

- فرآیند مهندسی نرم افزار

- روش منظم

- انتصاب وظایف و نقش ها

- توسعه RUP

- محدوده دانش

- نقش ها و فعالیت ها

- UMA (Unified Method Architecture)



مهندسی نرم افزار

- سیستم های امروزه
 - کنترل پرواز تعبیه شده
 - برنامه های کاربردی وب مبادله سهام
 - برنامه های کاربردی از راه دور راهنمایی دارویی
- قابلیت اتکا بدون هزینه سربار
- مهندسی نرم افزار
 - روش ها
 - تکنیک ها
 - ابزارها
 - نتایج کیفی



نرم افزار مطمئن

□ اطمینان در نرم افزار:

■ خطا

■ شناخت

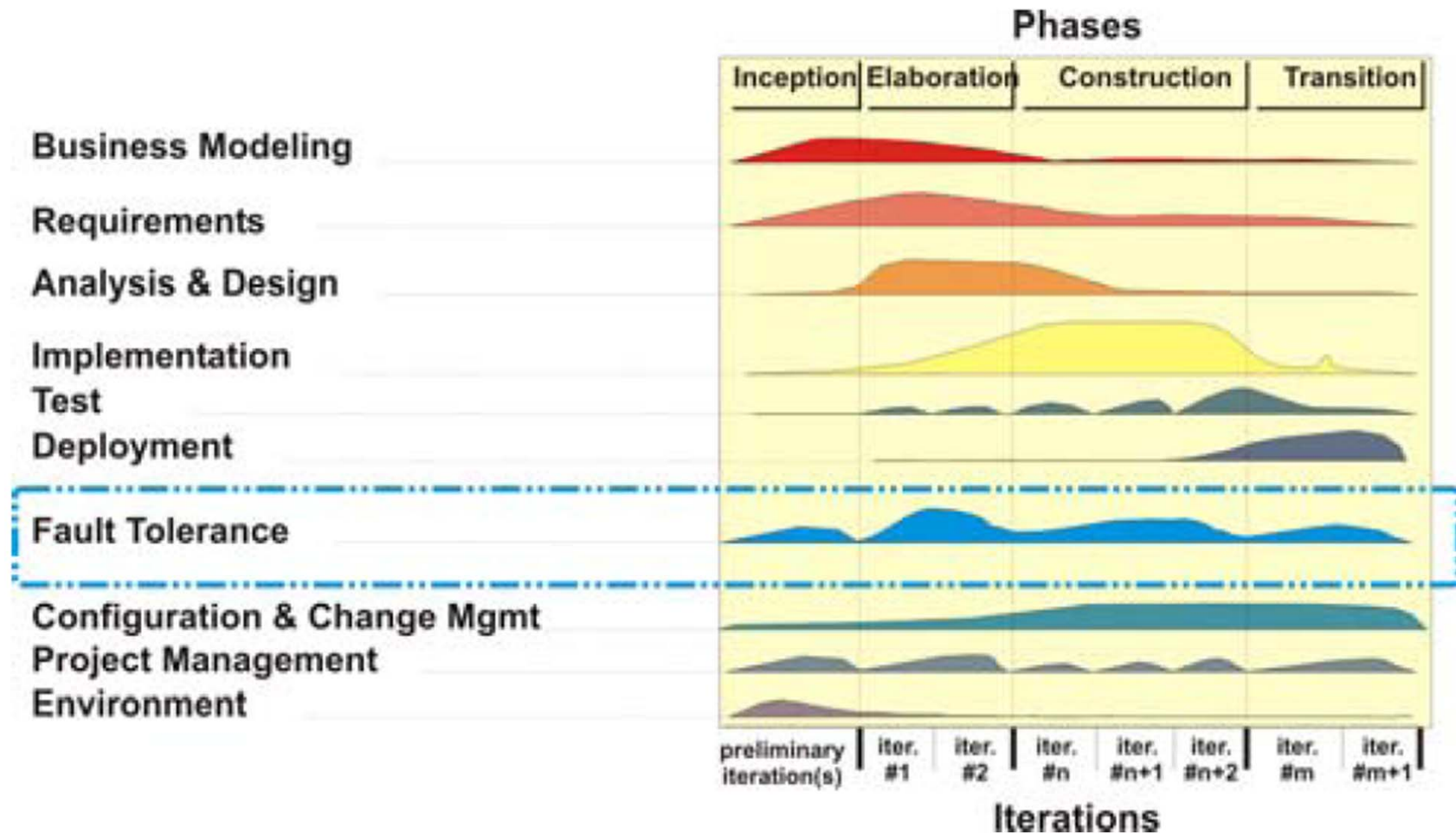
■ بازیابی

□ هدف:

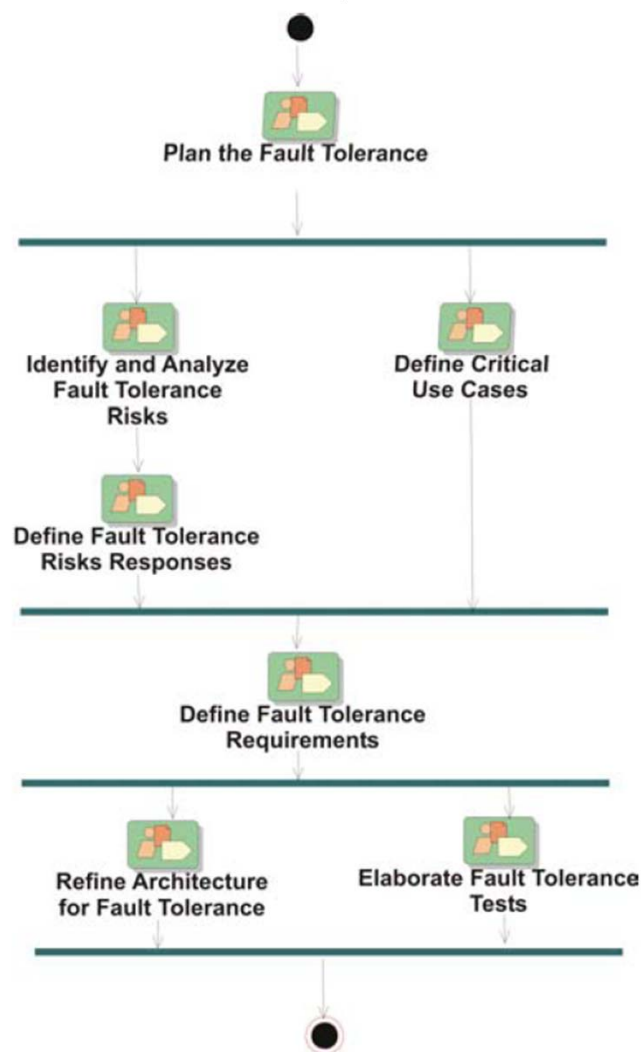
■ قابلیت اتکا

■ در دسترس بودن

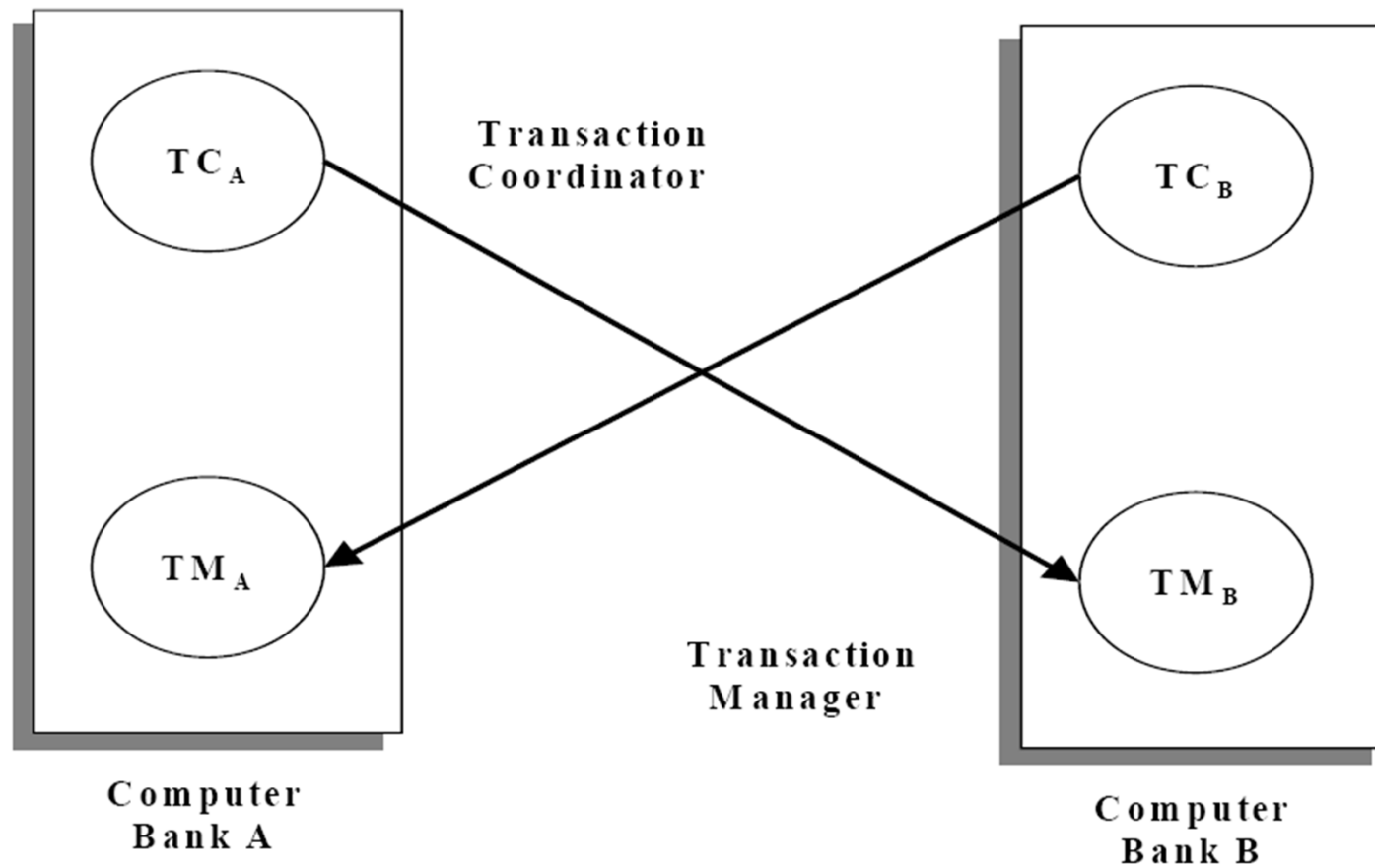
توسعه RUP



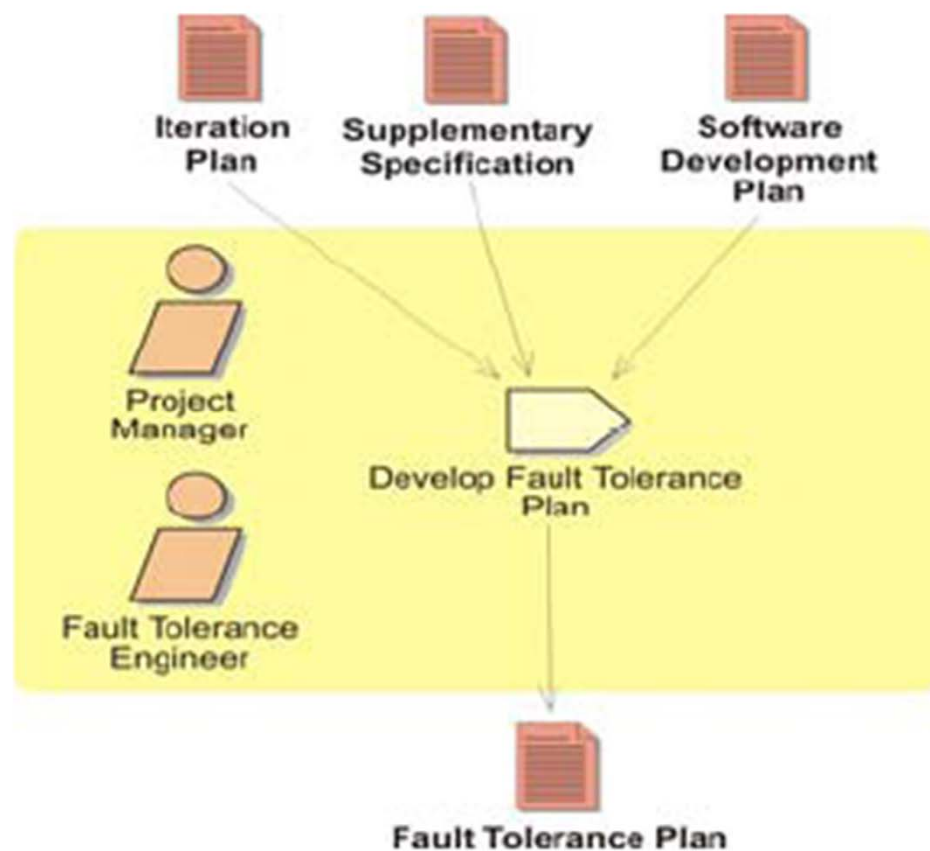
جریان کار روش منظم اطمینان در نرم افزار



مثال مورد بررسی



برنامه تحمل خطا

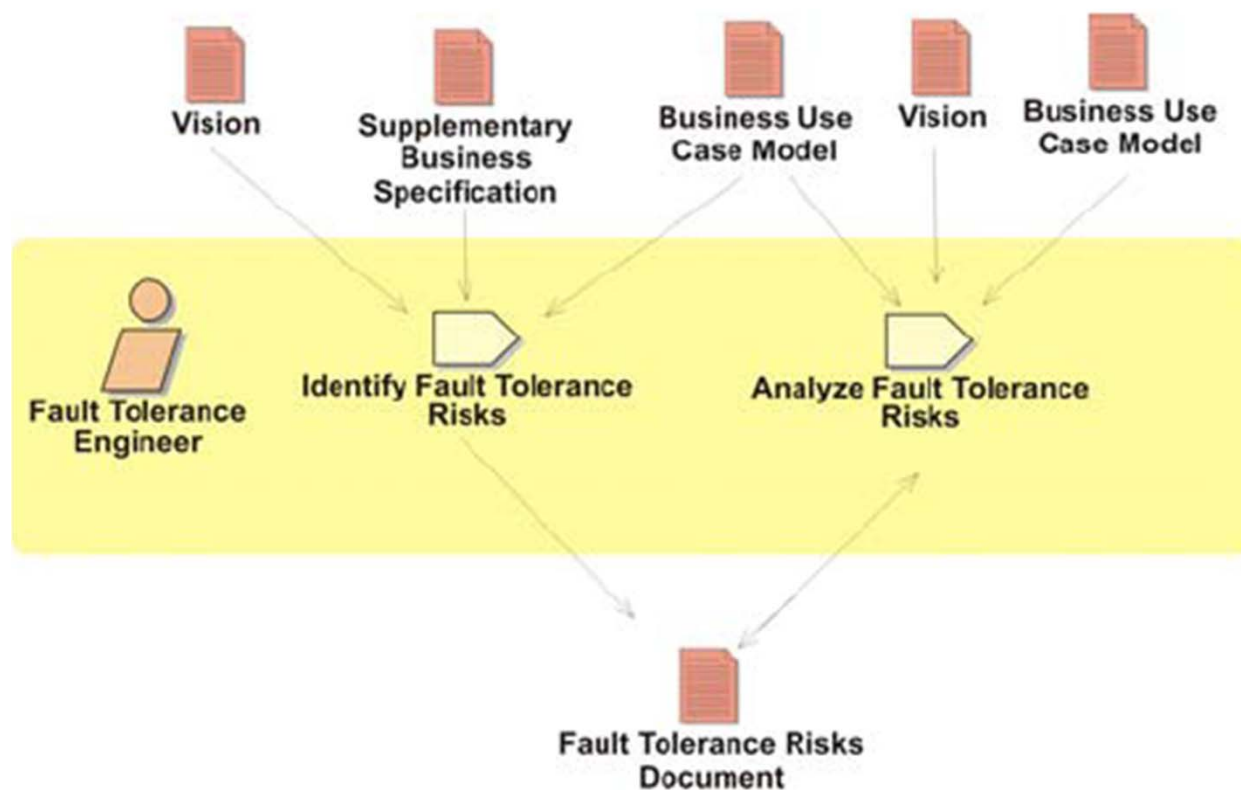


تحلیل درخت خطا

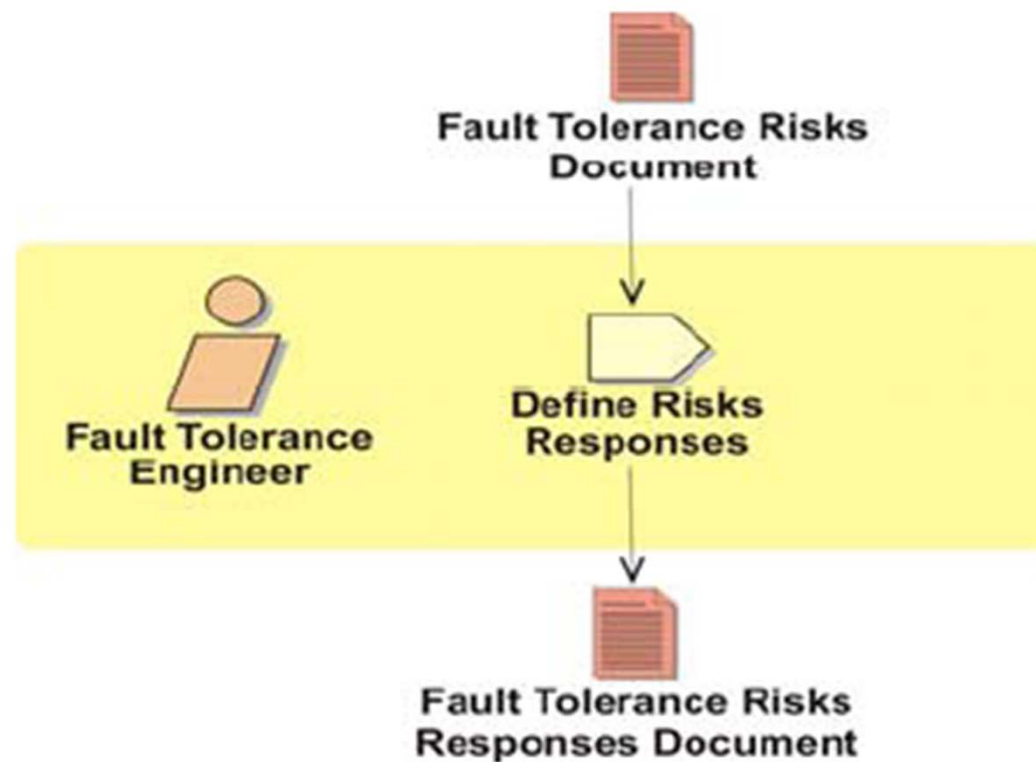
■ قدم ها:

- مشخص کردن رویدادهای نا مطلوب یا خطا از مورد های کاربرد حساس
- مرور جنبه های دیگر: محیطی، داده طراحی و تقاضای سیستم
- تعریف شرایط، رویدادهای خصوصی یا خطایی که با رویداد اصلی انتخاب شده ارتباط دارد.

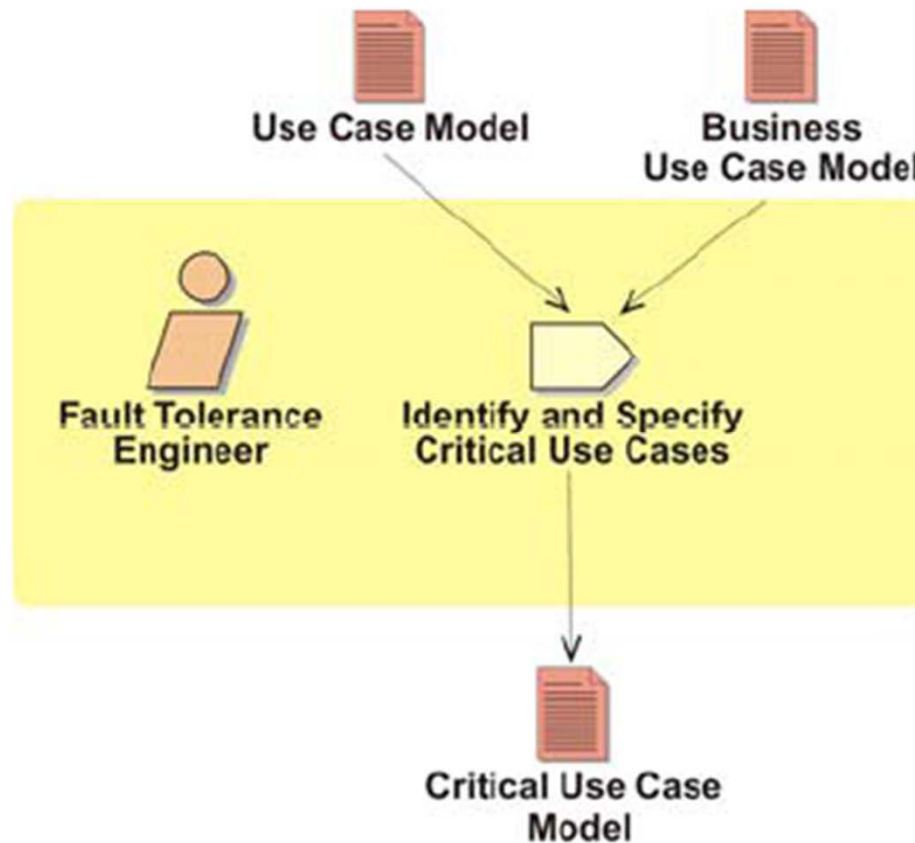
تشخیص و تحلیل ریسک های تحمل خطا



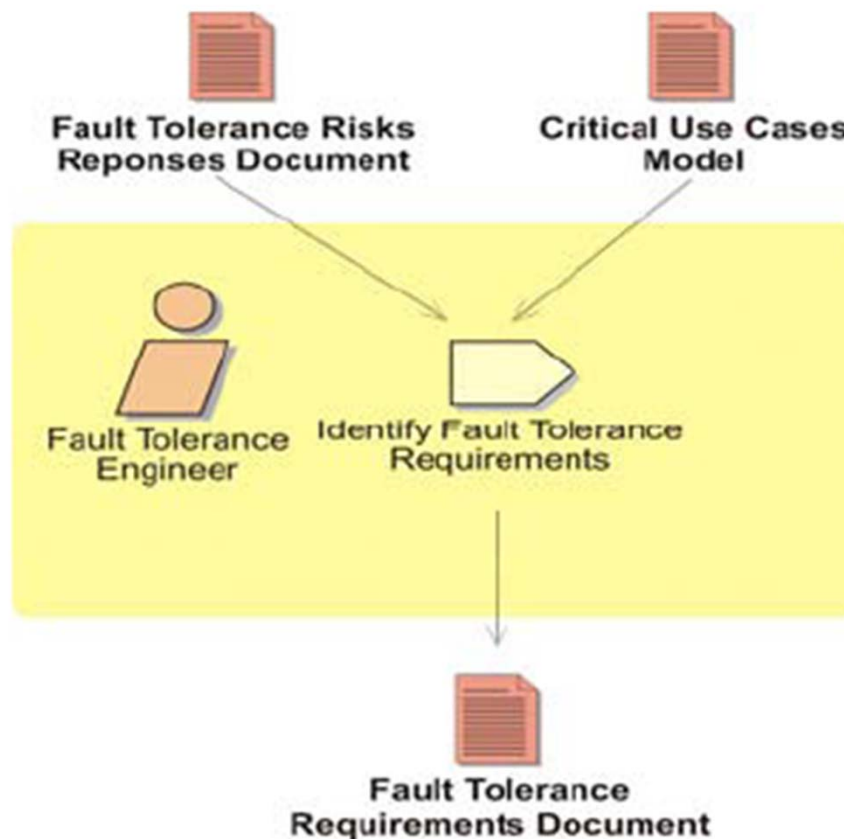
تعریف پاسخ های ریسک تحمل خطا



تعریف مورد های کاربرد حساس



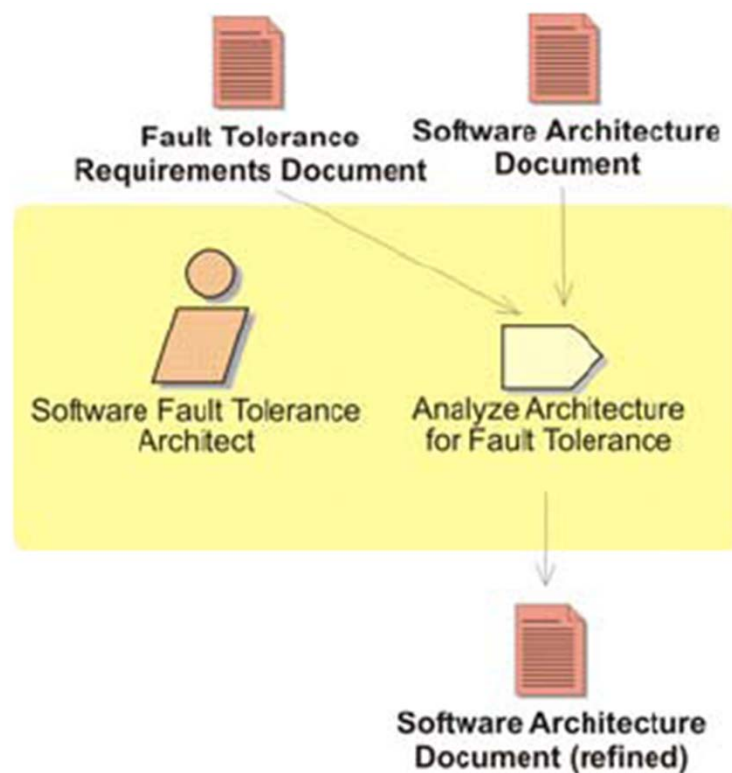
تعریف نیازمندی های تحمل خطا



مثال مستند نیازمندی های تحمل خطا

Fault Description	Server Crash
Target Component	Server Bank
Risk Impact	High
Recovery Mechanisms	Multiple versions of the server components, in other words, redundancy in two servers to improve availability.

تسفيه معماری برای تحمل خطا



تنظیم تست های تحمل خطا-1

■ هدف: مشخص کردن،

■ تست های تحمل خطای مورد نیاز

■ برنامه تست

■ تزریق تست های تحمل خطا

■ تکنیک تزریق خطا

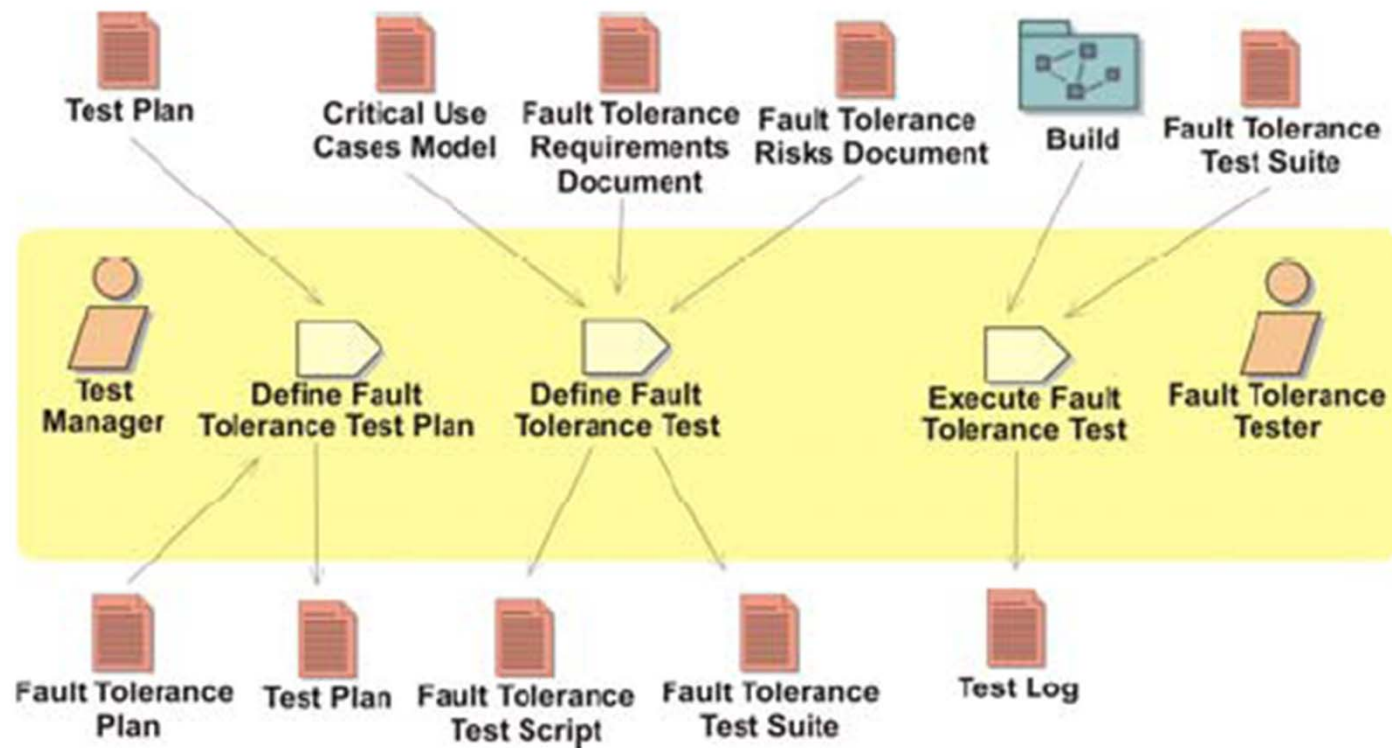
■ هدف: شبیه سازی تاثیر خطاهای حقیقی


□ انواع:

■ حین اجرا

■ حین کامپایل

تنظیم تست های تحمل خطا-2





مثال: ایجاد سیستم بانکی با استفاده از RUP

■ هدف اصلی ایجاد

■ فاز استخراج- inception

- مدل های مورد کاربرد،
- واژه نامه
- خصوصیات پشتیبانی: شامل نیازمندی های تحمل خطا
- برنامه تکرار

■ فاز تنظیم- Elaboration

- مدل های تحلیل و طراحی
- معماری سیستم
- مدل نصب
- مدل داده



ایجاد با استفاده از RUP متحمل خطا

- مدل موارد کاربرد حساس
- مستند ریسک
- مستند نیازمندی های تحمل خطا
- مستند معماری با دیدهای معماری مختلف از سیستم تسفیه شده برای تحمل خطا



نتیجه گیری

- مشخص کردن و تحلیل ساده تر و مستقیم تر نیازمندی های تحمل خطا.
- معماری سازگارتر و مناسب تر برای تحمل خطا.
- ارزیابی بهتر جنبه های امن بودن سیستم.
- زمان ایجاد نرم افزار کوتاه تر.
- 7 فعالیت، 10 وظیفه، 4 خروجی، 5 نقش اضافه.