

به نام خداوند بخشنده مهربان  
متن کتاب

۱۲ اسفند ۱۳۹۱



# فصل ۱

## متن کتاب

### ۱.۱ : توسعه توافقی نرم افزار<sup>۱</sup>

ASD که توسط James A. Highsmith توسعه یافت، یک روش انطباقی و چابک را برای پروژه‌های نرم‌افزاری با تغییرات بالا و سرعت بالا، پیشنهاد می‌دهد. انجام یک طرح‌ریزی موفق، با یک حرکت سریع و محیط کسب و کار غیر قابل پیش‌بینی، امکان‌پذیر نمی‌باشد. در ASD، چرخه حیات ایستای طراحی - برنامه‌ریزی با یک چرخه حیات پویای یادگیری - همکاری - تفکر، جایگزین می‌شود.

نقطه مرکزی ASD، سه فاز غیرخطی و دارای اشتراک با هم می‌باشد که عبارتند از:

- <sup>۲</sup>تفکر: برای تعریف کردن هیئت پروژه، فهم و درک خود را از هر چه که غیر مفهوم است، روشن کنید.
- <sup>۳</sup>همکاری: به معنی پررنگ نمودن نکات مهم تیم کاری به منظور توسعه سیستم‌های پرتغییر، می‌باشد.
- یادگیری: این فاز بر نیاز به پذیرش و واکنش نشان دادن به خطاها و این که نیازمندی‌ها ممکن است

---

<sup>۱</sup>ASD  
<sup>۲</sup>Speculate  
<sup>۳</sup>Collaborate

در طول فرآیند توسعه تغییر نمایند، تأکید می‌کند.

از آنجایی که خروجی‌ها طبیعتاً غیرقابل پیش‌بینی هستند، Highsmith فرآیند طرح‌ریزی<sup>۱</sup> را بعنوان یک تناقض در یک محیط انطباقی می‌بیند. در طرح‌ریزی سنتی، هنگامی که کارهای برنامه‌ریزی شده انجام نمی‌شوند، این امر به عنوان یک خطا حساب می‌شود که باید اصلاح شود. درحالی که در یک محیط توافقی، انحرافات، ما را به سمت راه‌حل درست، راهنمایی می‌کنند.

ASD روی نتایج و کیفیت‌های آنان بیشتر از وظایف یا فرآیندهایی که برای تولید نتایج استفاده می‌شوند، تمرکز می‌کنند. در یک محیط غیرقابل پیش‌بینی، شما به افرادی نیاز دارید تا با بکارگیری یک روش قطعی، به عدم قطعیت، رسیدگی نمایند. مدیریت بیشتر درباره تشویق نمودن ارتباطات است تا اینکه به افراد گفته شود که چکاری را باید انجام دهند و این کار به این دلیل است که جوابهای خلاقانه‌ای بدست آیند.

در محیط‌های کلاسیک قابل پیش‌بینی، طراحی‌ها از همان راهی پیروی می‌کنند که قبلاً داشته‌اند. بنابراین تشویقی برای یادگیری وجود ندارد. یادگیری یک ویژگی مهم و پیوسته است، چرا که فرض کرده‌اند که طرح‌ریزی و طراحی، به عنوان محصولات توسعه، باید تغییر کنند.

ASD مانند XP، اصول را جزئی نمی‌کند. اما یک فریمورکی را تهیه می‌کند تا تحت آن فریمورک ارتباطات تشویق شوند و یادگیری در طول پروژه وجود داشته باشد. ASD به عنوان یک متدولوژی برای ایجاد پروژه‌های نرم‌افزاری ارائه نشد، اما یک روش یا یک گرایش است که باید توسط سازمانی که فرآیندهای چابک را بکار می‌گیرد، پذیرفته شود.

## ۲.۱ درست

---

<sup>۱</sup>Planning

