

فهرست مطالب

| | |
|----|--|
| ۹ | آنچه قبل از شروع باید بدانید: |
| ۱۵ | فصل اول: انتخاب فرآیند و متدولوژی توسعه نرم افزار |
| ۱۵ | چرخه حیات توسعه نرم افزار یا فرایندهای توسعه نرم افزار |
| ۱۷ | SDLC چگونه کار می کند؟ |
| ۱۷ | مراحل SDLC و یک نمونه کاربردی از آن |
| ۱۸ | نمونه هایی از SDLC های عملیاتی |
| ۲۱ | مدل چابک (Agile) و متدولوژی اسکرام (Scrum) |
| ۲۹ | فصل دوم: اسکرام |
| ۲۹ | تعریف اسکرام (Scrum) |
| ۳۱ | نظریه اسکرام |
| ۳۱ | شفافیت |
| ۳۱ | بازرسی |
| ۳۲ | انطباق |
| ۳۲ | ارزش های اسکرام |
| ۳۲ | تیم اسکرام |
| ۳۳ | مالک محصول (Product Owner) |
| ۳۴ | تیم توسعه (Development Team) |
| ۳۶ | اسکرام مستر (Scrum Master) |
| ۳۷ | رویدادهای اسکرام (Scrum Events) |
| ۳۸ | اسپرینت (Sprint) |
| ۳۹ | لغو اسپرینت |

| | |
|----|--|
| ۳۹ | برنامه‌ریزی اسپرینت (Sprint Planning) |
| ۴۱ | هدف اسپرینت |
| ۴۲ | اسکرام روزانه (Daily Scrum) |
| ۴۳ | جلسه بررسی اسپرینت (Sprint Review) |
| ۴۵ | بررسی ساختاری اسپرینت (Sprint Retrospective) |
| ۴۶ | اشیا اسکرام (Scrum Artifacts) |
| ۴۶ | Product Backlog |
| ۴۸ | نظارت بر پیشرفت به سوی هدف |
| ۴۸ | Sprint Backlog |
| ۴۹ | نظارت بر پیشرفت اسپرینت |
| ۴۹ | توسعه |
| ۵۰ | شفافیت اشیا |
| ۵۰ | تعریف "کار انجام شده (Done)" |
| ۵۱ | سایر اشیا |
| ۵۳ | در پایان اسکرام |
| ۵۵ | فصل سوم: برآورد (Estimation) |
| ۵۷ | عناصر یک برآورد موفق |
| ۵۸ | مفروضات، برآوردها را دقیق‌تر می‌کند |
| ۶۰ | بی‌اعتمادی می‌تواند باعث تضعیف برآوردها شود |
| ۶۱ | اصول برآورد چابک |
| ۶۳ | اسکرام در رابطه با برآورد چه می‌گوید؟ |
| ۶۳ | موارد Product Backlog باید برآورد شوند |
| ۶۳ | افرادی که کار را انجام می‌دهند برآورد نهایی را می‌کنند |
| ۶۳ | برآورد آنچه می‌توان در اسپرینت انجام داد |
| ۶۳ | به روز رسانی برآورد در طول اسپرینت |
| ۶۵ | Story Point |
| ۶۷ | برنامه‌ریزی پوکر (Poker Planning) |
| ۶۹ | سیستم سطل (The Bucket System) |
| ۷۲ | روش دسته‌بندی بر اساس سایزها |
| ۷۴ | برآورد را متوقف کنید!!! (Stop Using Story Point) |

| | |
|-----|---|
| ۷۴ | مسیر اصلی روش‌های چابک |
| ۷۵ | بدون برآورد (NoEstimates)، یک روش نیست |
| ۷۷ | تیم‌هایی که برآورد انجام نمی‌دهند |
| ۸۱ | تفکر پروژه |
| ۸۳ | فصل چهارم: مستندسازی (Documentation) |
| ۸۶ | فواید و چرایی مستندسازی |
| ۹۲ | چه چیزهایی و چگونه باید مستند شوند؟ |
| ۹۶ | نیازمندی‌های نرم‌افزار (Software Requirements) |
| ۹۷ | استخراج نیازمندی‌ها |
| ۹۸ | انجام مصاحبه‌ها |
| ۱۰۱ | مشاهده کاربران در محل کار |
| ۱۰۲ | استفاده از خلاصه مباحث |
| ۱۰۳ | الگوی مستند خلاصه مباحث |
| ۱۰۶ | مشخصات نیازمندی‌های نرم‌افزار (Software Requirements Specification – SRS) |
| ۱۰۷ | یک قالب برای مستند SRS |
| ۱۱۳ | فصل پنجم: طراحی و پیاده‌سازی (Design & Implement) |
| ۱۱۵ | پیش به سوی UI/UX |
| ۱۱۵ | UI چیست؟ |
| ۱۱۷ | UX چیست؟ |
| ۱۱۸ | رابط کاربری چه تفاوتی با تجربه کاربری دارد؟ |
| ۱۲۱ | Usability |
| ۱۲۲ | تجربه مشتری (CX) |
| ۱۲۴ | استفاده از سورس کنترل |
| ۱۲۵ | قفل کردن در مقابل بازبودن |
| ۱۲۷ | سورس کنترل Subversion یا SVN |
| ۱۲۹ | سورس کنترل گیت (Git) |
| ۱۳۳ | یک کد خوانا و تمیز |
| ۱۳۴ | نام گذاری معنادار |
| ۱۳۶ | توابع |
| ۱۳۶ | توضیحات (Comments) |

| | |
|-----|--|
| ۱۴۳ | قالب‌بندی کد (Formatting) |
| ۱۴۴ | هدف از قالب‌بندی |
| ۱۴۷ | فصل ششم: تست نرم‌افزار (Software Testing) |
| ۱۴۸ | تست واحد (Unit Test) |
| ۱۴۷ | توسعه مبتنی بر تست (Test-Driven Development - TDD) |
| ۱۵۲ | همه در برابر کیفیت مسئول هستند |
| ۱۵۴ | انجام تست واحد زمان برنامه‌نویسی را کاهش می‌دهد |
| ۱۵۵ | برنامه‌ریزی‌های تست و موارد تست (Test Planes & Test Cases) |
| ۱۵۹ | اجرای تست |
| ۱۶۱ | ردیابی نقایص و خطاها و اولویت‌بندی آنها |
| ۱۶۳ | محیط تست و تست عملکرد |
| ۱۶۸ | تست‌های خودکار (Test Automation) |
| ۱۷۰ | استفاده مؤثر از تست نرم‌افزار |
| ۱۷۱ | تسترها چه کاری انجام می‌دهند |
| ۱۷۴ | تقسیم وظایف مربوط به کیفیت به طور مؤثر |
| ۱۷۶ | مدیریت فشارهای زمانی |
| ۱۷۷ | تشخیص مشکلات تست نرم‌افزار |
| ۱۷۸ | نیازمندی‌هایی که پیاده‌سازی نشده‌اند |
| ۱۷۹ | بروز خطاهای واضح |
| ۱۸۳ | فصل آخر: توسعه عشق |