

A field study of API learning obstacles

Empirical Software Engineering

Martin P. Robillard e Robert DeLine

Hudson Silva Borges

10 de Setembro de 2012

Introdução

- Aprender a usar APIs é difícil;
- Poucos estudos focam em descobrir fatores que mais causam isto;
- Várias pesquisas foram realizadas, mas objetivavam obter as respostas mínimas necessárias;

Introdução

- Objetivo do estudo:
 - Determinar de forma mais abrangente e específica quais são os obstáculos enfrentados pelos desenvolvedores quando aprendem uma nova API;
- Metodologia:
 - Pesquisa exploratória;
 - Entrevistas qualitativas;
 - Pesquisa de acompanhamento;

Trabalhos Relacionados

- Trabalhos pioneiros;
- Informações necessárias aos desenvolvedores;
- Estudos sobre Usabilidade;
- Estudos sobre documentação;
- Experiência prática documentada;

Visão Geral da Pesquisa

- Abordagem:
 - Multi-fase;
 - Métodos mistos;
 - Sucessivos refinamentos;
- # 1 – Determinar o que faz as APIs serem difíceis de aprender;
- # 2 – Detalhar as dificuldades;
- # 3 – Provar a hipótese que “Documentação adequada é o obstáculo mais severo no aprendizado”

População Alvo

- Equipe de desenvolvimento da Microsoft;
- População de 30.000 engenheiros entre desenvolvedores, testadores e gestores;
- Testadores não foram incluídos;
- Distinção de cargos e níveis;

Pesquisa Exploratória

- Questões principais abertas;
- Citar três obstáculos;
- Pesquisa enviada a 1000 funcionários da Microsoft (Redmond) selecionados randomicamente;
- Não foram oferecidos nenhum tipo de prêmio por respostas;

Pesquisa Exploratória

- 80 respostas obtidas (8%);
- 12,9 anos de experiência de desenvolvimento em média;
- Conhecimento sobre 54 APIs em média;
- 165 respostas sobre obstáculos foram obtidas;
- Análise das respostas usando “palavras-chave”

Pesquisa Exploratória

Table 1 Obstacle categories (exploratory survey)

Obstacle related to...	# resp.
Learning resources (e.g., documentation, code examples)	50
API structure (e.g., design, names of API elements)	36
Developer background (e.g., prior knowledge, professional training)	17
Technical environment (e.g., build tools, testing infrastructure, hardware)	15
Process (e.g., lack of time, interruptions by colleagues, other priorities)	13

Entrevista Qualitativa

- Objetivo de obter maiores detalhes, descobrir contextos que ocorrem, e inferir possíveis causas;
- 28 participantes dos 80 da fase anterior, 2 gerentes de software e 1 chefe de desenvolvimento;
- Duração entre 15 e 45 minutos;
- Quando um tema se sobressaia era abordado nas entrevistas posteriores;

Entrevista Qualitativa

Table 3 Learning contexts for participants

Context	Learning the API...
Owning	because the respondent joined the team developing it
Major	to use as major component of production code
Minor	to complete some minor task (e.g., bug fix)
Exp.	to experiment with a technology
Hobby	for a side project not critical to main work

Pesquisa de acompanhamento

- Objetivo de tentar ligar as características do desenvolvedor aos obstáculos;
- Envio a 2000 funcionários da Microsoft (excluindo participantes anteriores);
- Prêmio de 250 dólares na forma de cartão de presente em uma loja online;
- 334 (17,3%) desenvolvedores responderam;

Pesquisa de acompanhamento

- Primeira e segunda questão relacionada à última experiência com APIs;
- Terceira questão relacionada ao grau de dificuldade de obstáculos de cinco perspectiva:
 - Background, Structure, Technical Environment, Low-level documentation e High-level documentation;
- Avaliar 14 soluções derivadas da literatura e das pesquisas;

Resultados Cuantitativos

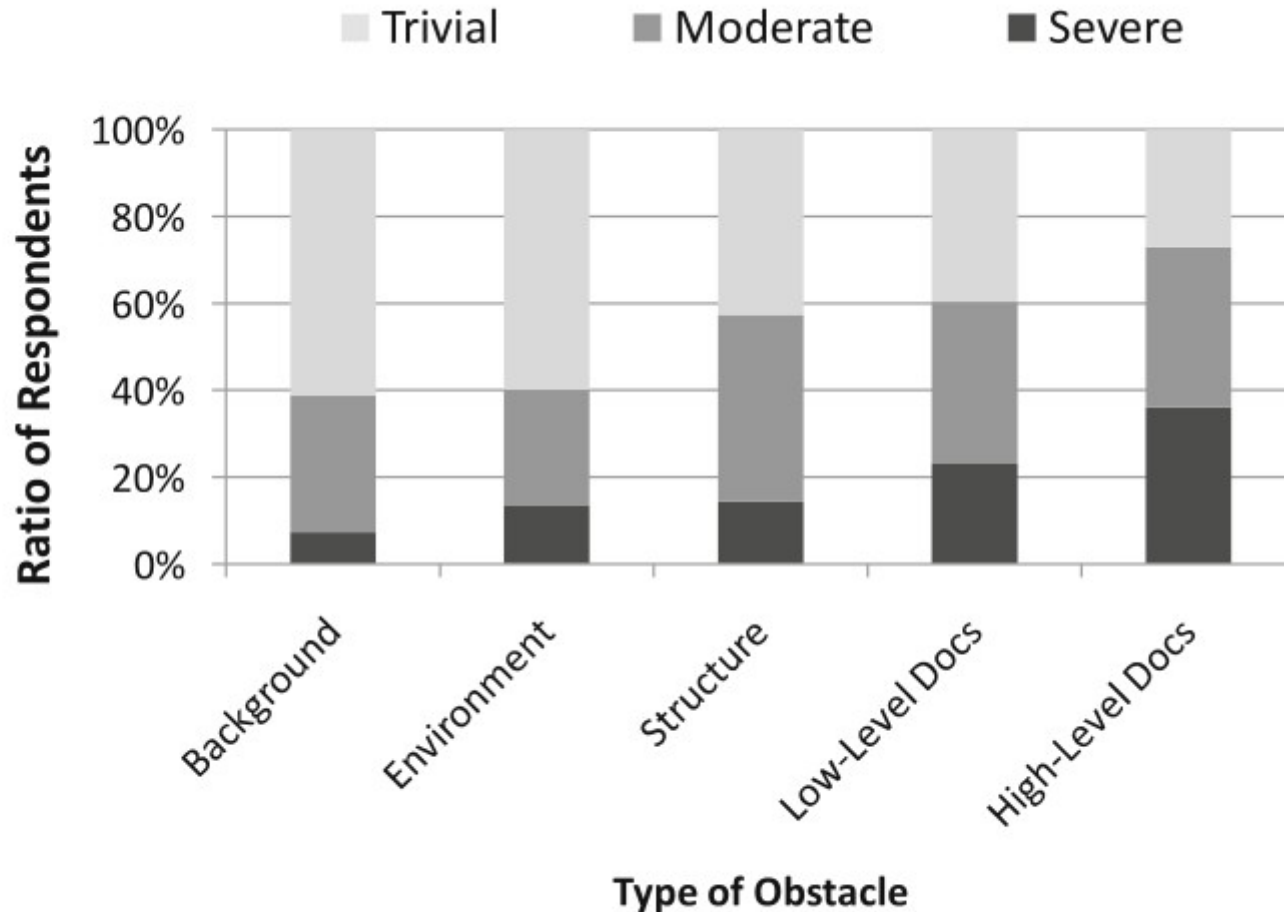


Fig. 1 Obstacle severity (follow-up survey)

Resultados Cuantitativos

Relation / Relative severity	<	>	=
Background ? Structure	151	56	124
Background ? Environment	116	91	124
Background ? Low-level docs	169	78	80
Background ? High-level docs	208	39	83
Structure ? Environment	76	143	112
Structure ? Low-level docs	131	87	109
Structure ? High-level docs	158	57	115
Environment ? Low-level docs	160	78	89
Environment ? High-level docs	195	55	80
Low-level Docs ? High-level docs	135	67	124

Resultados Qualitativos

- Documentação inadequada é o maior obstáculo;
- Ainda assim é uma área muito ampla, então é necessário descobrir os aspectos que mais influenciam;
- A partir das etapas anteriores foram elicitadas cinco dimensões;
- Para cada análise algumas recomendações de priorização foram apontadas;

Resultados Qualitativos

- Para apoiar as descobertas, evidências obtidas nas entrevistas, formulários e literatura foram utilizadas;
- Foco em explicar alguns fatores humanos envolvidos nos fenômenos, não na frequência que ocorrem;

Dimensão 1 – Documentação do Objetivo

- Prover importantes informações sobre decisões de projeto;
- Dos entrevistados, pelo menos oito apresentaram algumas vantagens;
- Algumas são:
 - Raciocínio sobre a maneira correta e eficiente de usar;
 - Determinar como implementar códigos avançados;
 - Argumentar sobre características de performasse;

Dimensão 1 – Documentação do Objetivo

- **Implicação 1**

- A documentação do objetivo deve ser fornecido de acordo com a necessidade. Sessões onde o uso correto não é evidente, para apoiar uso avançado de códigos, e aspectos de desempenho são algumas das possíveis áreas beneficiadas;

Dimensão 2 – Exemplos de código

- Exemplos de código são listagens de vários tamanhos que mostram o uso de APIs;
- Distinção em três principais características:
 - Curto trecho de código (aspectos específicos)
 - Sequências de pequenos trechos de código (pequeno mas não trivial funcionalidade)
 - Pequenas aplicações (pequenos mas auto-contidos)
 - Produtor de código (fontes de pesquisa de código)

Dimensão 2 – Exemplos de código

- Podem ser provido de contextos;
- Diversos estudos apontam repetidamente que exemplos são elementos essenciais;
- Mas quão complexo pode ser?
 - Relação do tamanho com poder pedagógico;
 - Trechos pequenos com um método tem pouco valor;
- Combinação métodos em geral é mais útil;

Dimensão 2 – Exemplos de código

- **Implicação 2**

- Pequenos exemplos que demonstram um padrão de uso da API e envolve mais de um método pode ser mais útil que chamadas únicas;

Dimensão 2 – Exemplos de código

- Exemplos como recomendações;
- Exemplos como demonstração:
 - "Assim é como você deve usar a API"
- Exemplos como recomendação:
 - "Assim é como você poderia usar a API"
- Guia de “boas práticas”;

Dimensão 2 – Exemplos de código

- **Implicação 3**

- Exemplos poderiam ser desenvolvidos para demonstrar "boas práticas" para uso de APIs;

Dimensão 3 – Combinar APIs com cenários

- Uma dificuldade central na aprendizagem de APIs é descobrir os elementos da API adequados para o cenário;
- Um exemplo é a existência de diversos construtores de uma classe, qual seria o mais adequado?

Dimensão 3 – Combinar APIs com cenários

- **Implicação 4**

- Combinar cenários com elementos de APIs é uma área onde a documentação de suporte pode ser muito útil, e compensaria a difícil compreensão da estrutura.

Dimensão 4 – Penetrabilidade

- Uma das 12 dimensões cognitivas de Clarke para avaliação da usabilidade de APIs;
- Definição: "Como a API facilita a exploração, análise e compreensão do componente";
- Vários participantes relataram que em algum momento precisaram conhecer como determinadas API funcionam internamente;

Dimensão 4 – Penetrabilidade

- **Implicação 5**

- Documentação explícita sobre aspectos de desempenho de métodos, melhor descrição para tratamento de erros, descrição de operações abstratas da API, são exemplos de potenciais meios para reduzir as inspeções no código e prototipagens necessárias para aprendizado de APIs;

Dimensão 5 – Formato de documentação

- *"Some [documentation] pages talk about one method or have very little additional information and waste your time navigating through them. [Survey]"*

Dimensão 5 – Formato de documentação

- **Implicação 6**

- *Navegar pela documentação é uma consulta implícita. Documentações padrão frequentemente não respondem à consulta e desperdiçam o tempo do desenvolvedor;*

Dimensão 5 – Formato de documentação

- Várias das recomendações ressaltam a importância da documentação de nível conceitual, mas será o ideal ou a melhor forma?
- Documentação eletrônica é comumente fragmentada em artigos na internet;
- Participantes foram fortemente a favor de apresentações relativamente contínuas e não fragmentadas;

Dimensão 5 – Formato de documentação

- Implicação 7
 - Desenvolvedores procuram por apresentações coerentes e lineares, coleções de artigos fragmentados através de hyperlinks podem ser "esmagadores".

Críticas ao Experimento

- O uso de múltiplas fonte ajuda a obter um retrato fiel dos fenômenos de interesse;
- Pesquisa exploratória e de acompanhamento baseada em uma verdadeira amostra randômica de uma população conhecida;
- As entrevistas seguiram um caráter qualitativo, e é sujeito a interpretações diferentes;
- Pesquisa realizada em uma única organização;

Conclusões

- Desenvolvedores não focam tanto para a mecânica da API, mas no entendimento do domínio do problema (ex: nível conceitual e de classe);
- 7 implicações para priorização na documentação baseadas em 5 dimensões foram apresentadas;
- Assim o trabalho pode ser usado para coordenar a equipe de projeto com a de documentação;

Dúvidas?

Obrigado!
