Understanding Developers' Needs on Deprecation as a Language Feature

Anand Ashok Sawant, Maurício Aniche,
Arie van Deursen and Alberto Bacchelli

TU Delft

University of Zurich[UZH]

Understanding Developers' Needs on Deprecation as a Language Feature

Anand Ashok Sawant, Maurício Aniche,
Arie van Deursen and Alberto Bacchelli

TU Delft

University of Zurich[UZH]

What is deprecation (in Java)?

A program element annotated @Deprecated is one that programmers are discouraged from using, typically because it is dangerous, or because a better alternative exists. – **Java documentation on deprecation**

What is deprecation (in Java)?

A program element annotated @Deprecated is one that programmers are discouraged from using, typically because it is dangerous, or because a better alternative exists. – **Java documentation on deprecation**

What is deprecation (in Java)?

A program element annotated @Deprecated is one that programmers are discouraged from using, typically because it is dangerous, or because a better alternative exists. – **Java documentation on deprecation**

What is deprecation (in Java)?

A program element annotated @Deprecated is one that programmers are discouraged from using, typically because it is dangerous, or because a better alternative exists. – **Java documentation on deprecation**

What is deprecation (in Java)?

A program element annotated @Deprecated is one that programmers are discouraged from using, typically because it is dangerous, or because a better alternative exists. – **Java documentation on deprecation**

Why study deprecation?

Why study deprecation?


```
import java.sql.Date;

public class Demo {
    public static void main(String args[]) {
        Date date = new Date(1L);
        date.set|;
    }

}
```

setHours(int i) : void - Date - used
setDate(int date) : void - Date
setMinutes(int i) : void - Date
setMonth(int month) : void - Date
setSeconds(int i) : void - Date
setTime(long date) : void - Date
setYear(int year) : void - Date

Press '^Space' to show Template Proposals

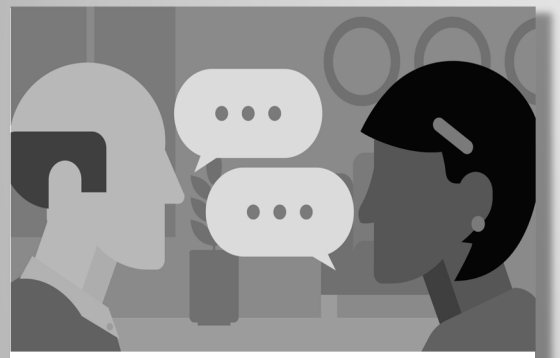
Deprecated.

This method is deprecated and should not be used because SQL Date values do not have a time component.

Overrides: setHours(...) in Date
Parameters:
i the hour value.
Throws:
java.lang.IllegalArgumentException - if this method is invoked
See Also:
getHours

Press 'Tab' from proposal table or click for focus

Why study deprecation?



Communication method

Changes often

OpenJDK

OpenJDK FAQ
Installing
Contributing
Sponsoring
Developers' Guide

Mailing lists
IRC · Wiki

Bylaws · Census
Legal

JEP Process

search

Source code
Mercurial
Bundles (6)

Groups
(overview)
2D Graphics
Adoption
AWT
Build
Compatibility & Specification Review
Compiler
Conformance
Core Libraries
Governing Board
HotSpot
Internationalization
JMX
Members
Networking
NetBeans Projects
Porters
Quality
Security
Serviceability
Sound
Swing
Web

Projects
(overview)
Amber
Annotations Pipeline 2.0
Audio Engine
Build Infrastructure
Caciocavallo
Closures
Code Tools

JEP 277: Enhanced Deprecation

Owner	Stuart Marks
Created	2014/11/20 23:58
Updated	2017/12/08 23:29
Type	Feature
Status	Closed / Delivered
Component	core-libs / java.lang
Scope	SE
Discussion	jdk9 dash dev at openjdk dot java dot net
Effort	M
Duration	M
Priority	2
Reviewed by	Alex Buckley, Mark Reinhold
Endorsed by	Brian Goetz
Release	9
Issue	8065614

Summary

Revamp the `@Deprecated` annotation, and provide tools to strengthen the API life cycle.

Goals

- Provide better information about the status and intended disposition of APIs in the specification.
- Provide a tool to analyze an application's static usage of deprecated APIs.

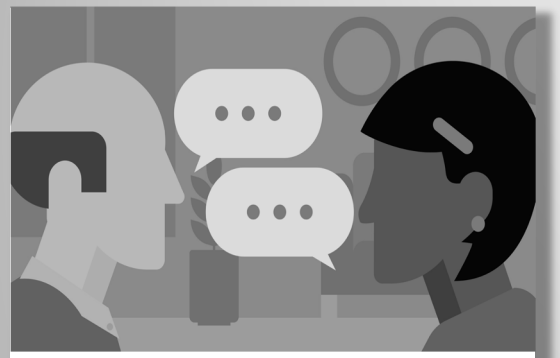
Non-Goals

It is not a goal of this project to unify the `@deprecated` Javadoc tag with the `@Deprecated` annotation.

Motivation

Deprecation is a technique to communicate information about the life cycle of an API: to encourage applications to migrate away from the API, to discourage applications from forming new dependencies on the API, and to inform developers of the risks of continuing dependence upon the API.

Why study deprecation?



Communication method

Changes often

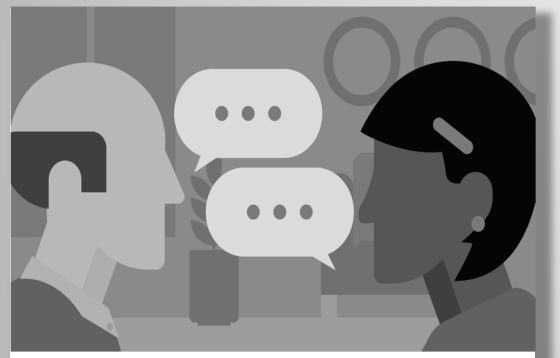
`forRemoval()`

Indicates that a deprecated method is going to be removed

`since()`

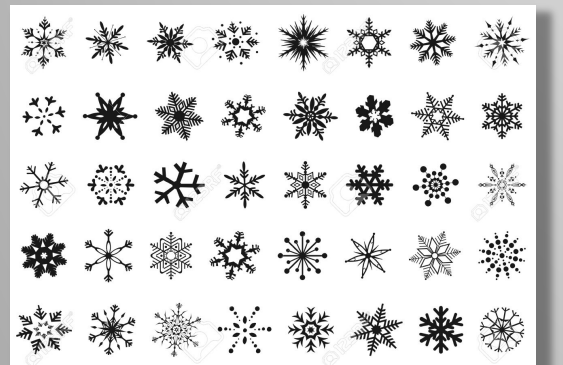
Indicates when the method was deprecated

Why study deprecation?



Communication method

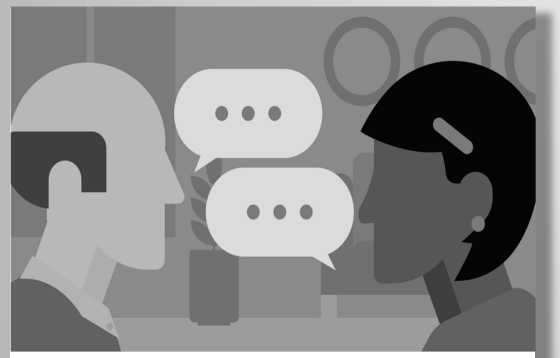
Changes often



No single implementation

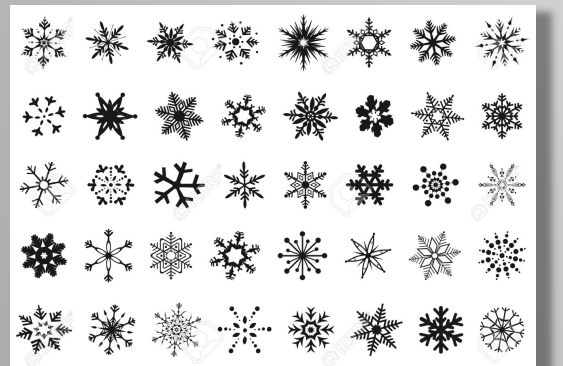
```
/**
  * @deprecated Use getIntValue instead.
  */
@Deprecated
 public int getValue() {
     return this.value;
}
```

Why study deprecation?



Communication method

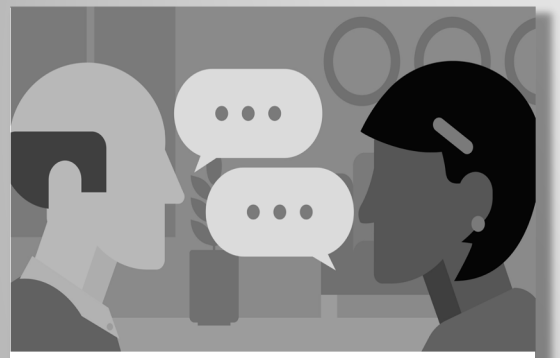
Changes often



No single implementation

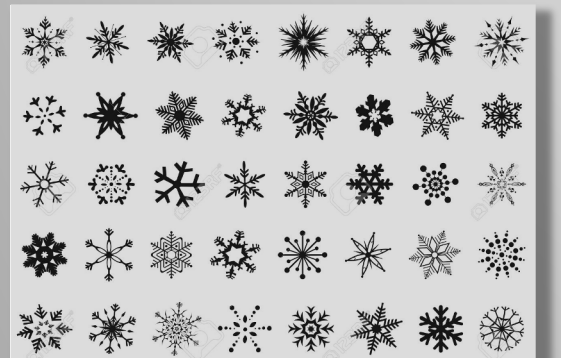
```
/**
  * @deprecated
  * @return $this
  */
public function oldMethod() {
    trigger_error('Method ' . __METHOD__ . ‘
is deprecated', E_USER_DEPRECATED);
    return $this;
}
```

Why study deprecation?



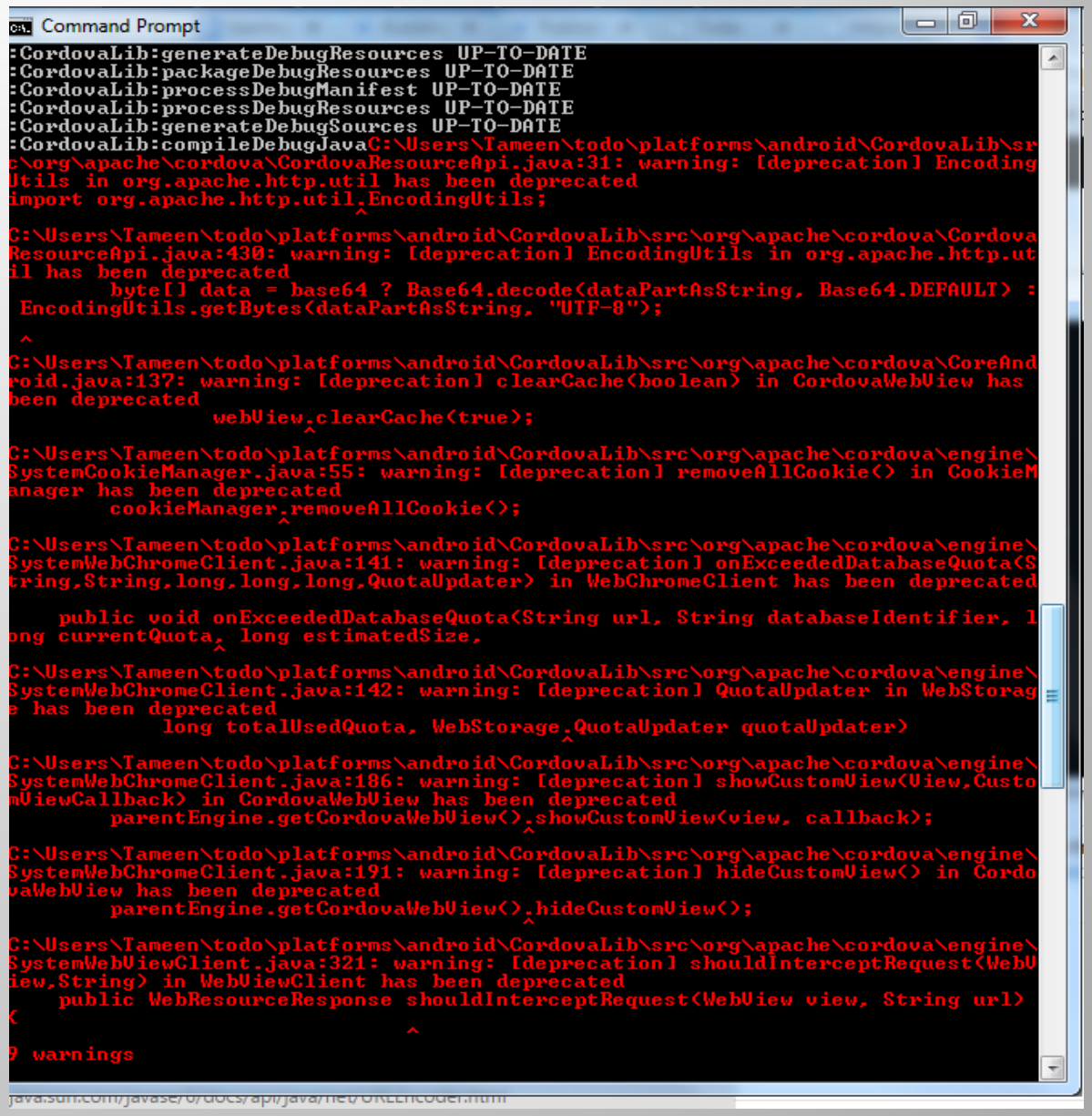
Communication method

Changes often



No single implementation

Not taken seriously



Our study

1

Investigate needs of API producers and consumers

Our study

Investigate needs of API producers and consumers

1

Enhancements to deprecation

2

Research questions

Investigate needs of API producers and consumers

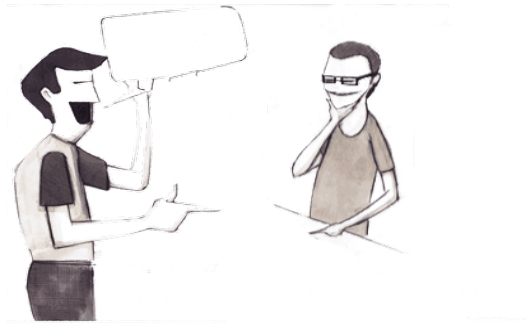
RQ1: Why do API producers deprecate features?

RQ2: When and why are deprecated features removed?

RQ3: How is an API consumer expected to react to deprecation?

RQ4: Why do API consumers react to deprecated features?

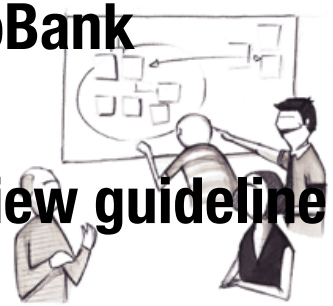
Methodology



Interview 17 API producers

OSS projects: Junit, Spring, Mockito

Industry: HP, ABN Amro, ING and RaboBank



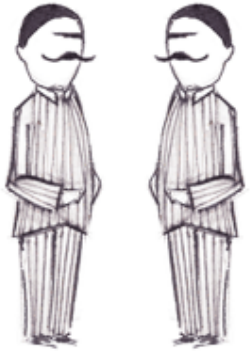
Emergent themes

Adapt interview guideline

Analysis of transcripts

Survey API consumers

170 survey respondents



Survey API producers

Emergent themes

1

Emergent themes

Future

Prefer maintaining two versions of the same feature in the code base and only introduce breaking changes when there is a severe flaw in the feature **– P12**

Consumers do not appreciate it when functionalities are removed **– P8**

Emergent themes

Future

Future of the deprecated feature is unknown so there is no need to react **– Survey respondent**

1

Emergent themes

Future

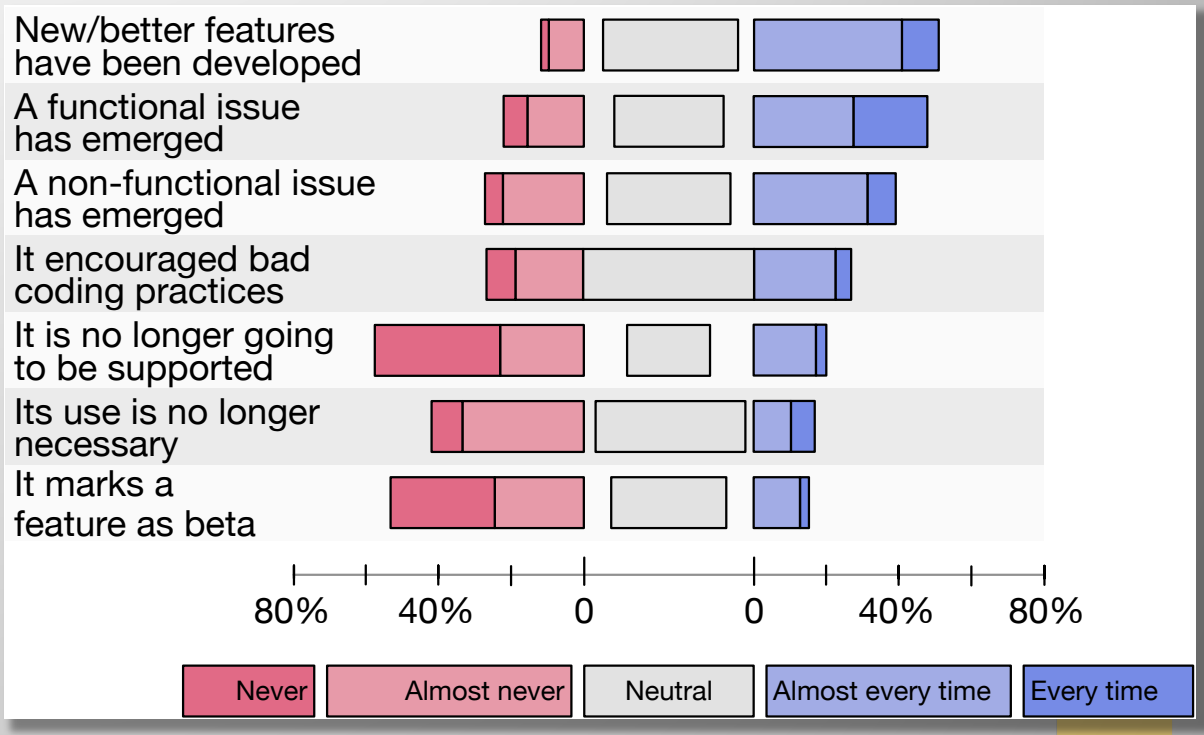
Severity

The decision to react is often based on the cost of upgrade. If the reason behind deprecation is serious, then there is an incentive to change **– P2**

Emergent themes

Future

Severity



Emergent themes

Future

Severity

Misuse

Deprecation is used to indicate that a feature is temporary and may change **– P16**

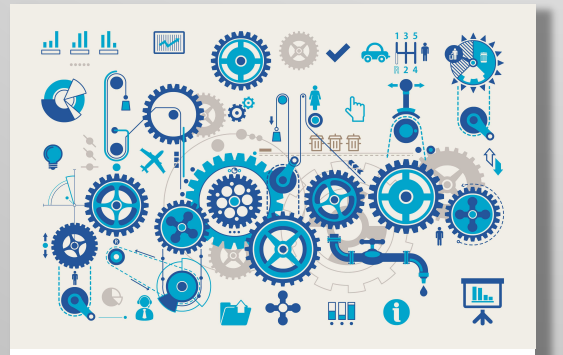
Instead of misusing deprecation a new annotation (@Unstable) was introduced to indicate beta features **– P14**

Emergent themes

Future

Severity

Misuse



Automation

An automated tool that aids consumers in transitioning from a deprecated feature to its replacements would be very helpful **– All interviewees**

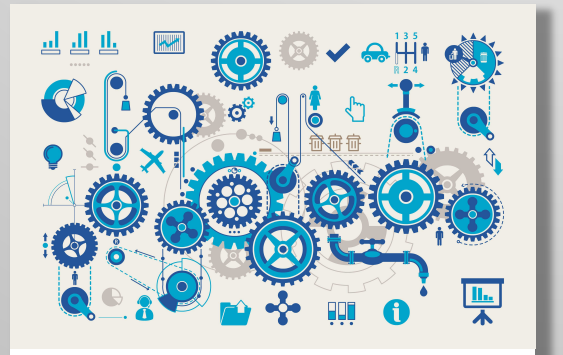
1

Emergent themes

Future

Severity

Misuse



Automation

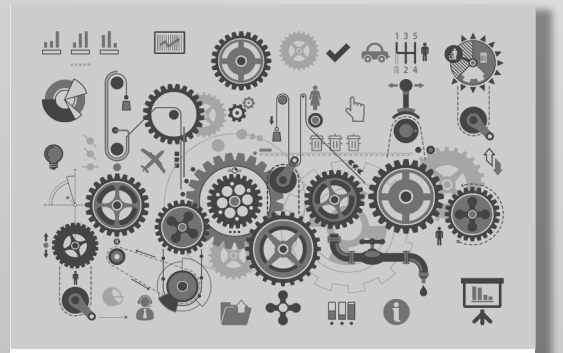
Investigate needs of API producers and consumers

RSW proposal

Future

Severity

Misuse



Automation

Enhancements to deprecation

2

RSW proposal

Future

Severity

Misuse

Enhancements to deprecation

2

RSW proposal

Future

Severity

Misuse

```
@Deprecated(
removalVersion="2.2.4")
```

```
@Deprecated(
removalDate="24-03-18")
```

RSW proposal


```
@Deprecated(
severity=Severity.LOW)
```

```
@Deprecated(level = 1)
@Deprecated(level = 2)
```

RSW proposal

R Removal timeline

Future

S Indicate severity

Severity

W Generic warning

Misuse

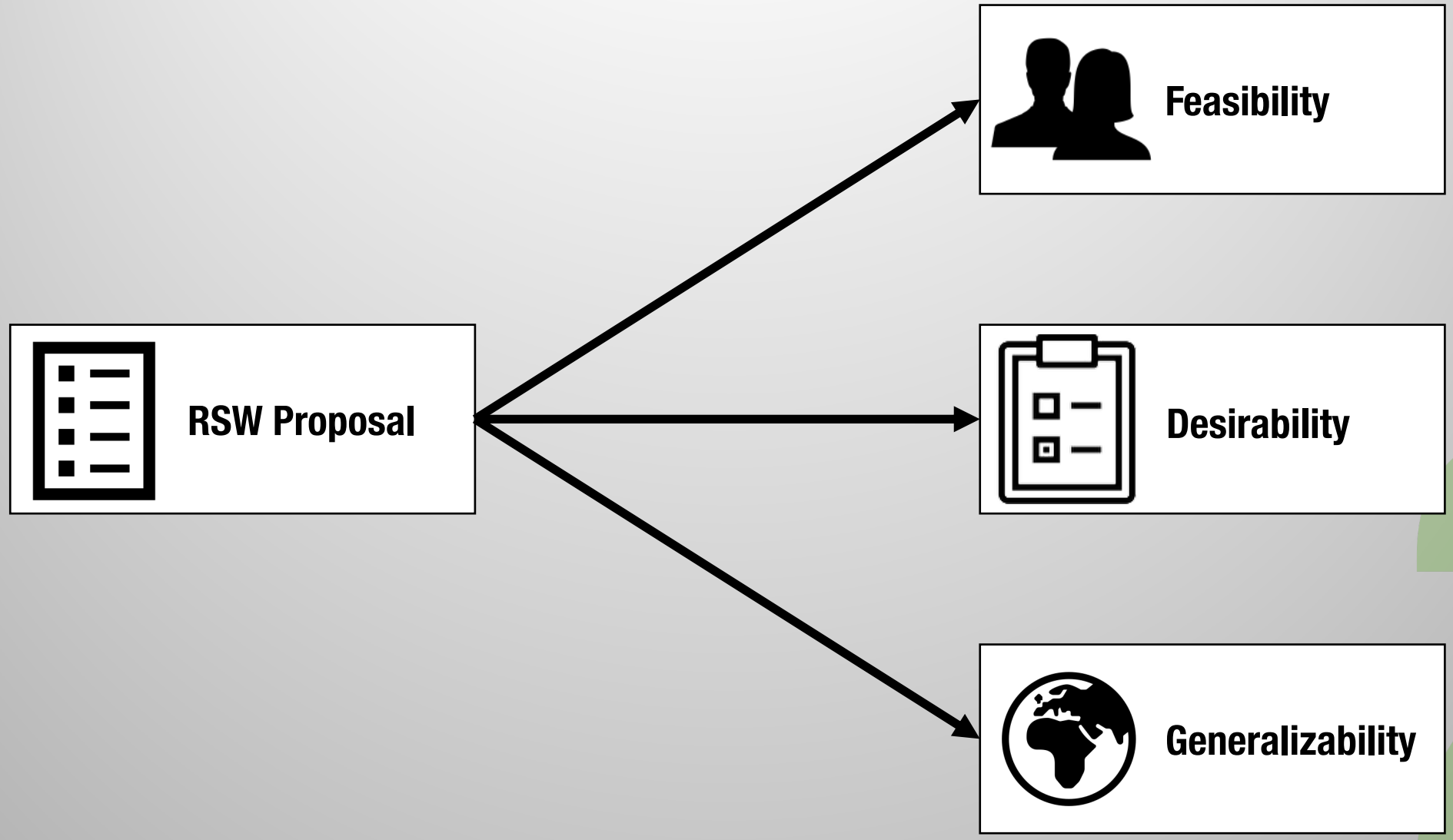
```
@Warning

  @Deprecated

  @Beta("Beta feature")
```

2

Evaluation methodology of the RSW proposal



Evaluation of the RSW proposal

Stuart Marks
Java Language Designer
a.k.a DrDeprecator

R Possible, not trivial
Challenge: Version or date?

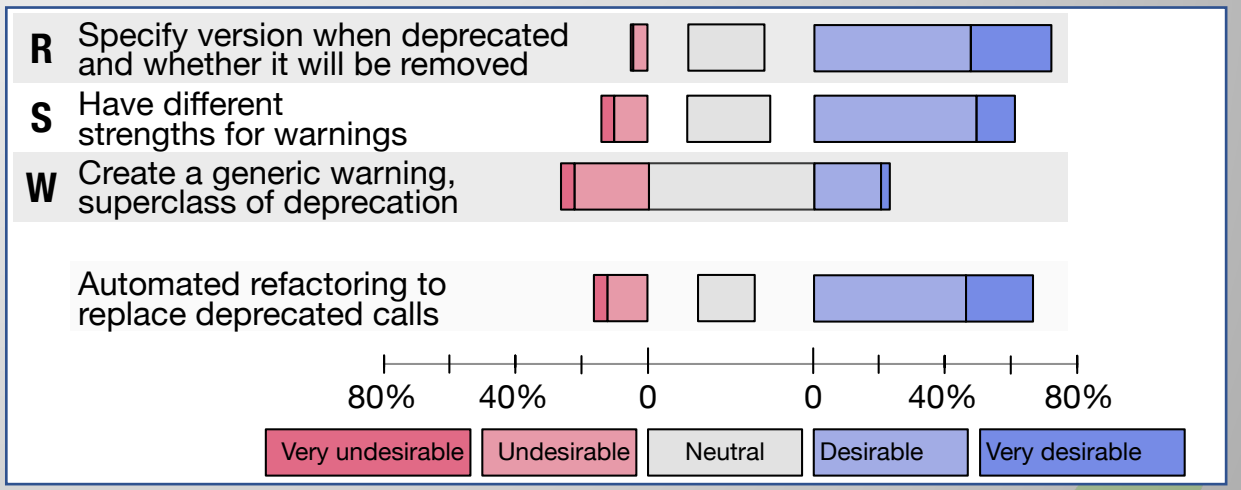
S Possible, not trivial
Challenge: defining severity

W Very complex
Extreme solution

Evaluation of the RSW proposal

Feasibility

Desirability



Evaluation of the RSW proposal

Feasibility

Desirability

Generalizability

R Ruby, Dart

S C#, VBasic, Kotlin

W PHP, Ruby, Delphi, R, ObjectiveC, Hack

2

Evaluation of the RSW proposal

Feasibility

Desirability

Generalizability

R Ruby, Dart

S C#, VBasic, Kotlin

W PHP, Ruby, Delphi, R, ObjectiveC, Hack

2

Understanding Developers' Needs on Deprecation as a Language Feature

Our study

Investigate needs of API producers and consumers

1

Enhancements to deprecation

2

Emergent themes

Investigate needs of API producers and consumers

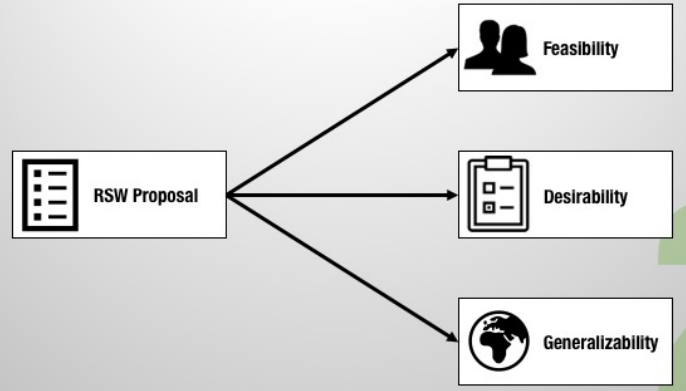
1

RSW proposal

Enhancements to deprecation

2

Evaluation methodology of the RSW proposal



2