RED•pill
embrace reality

Entwickler Fitness-Check
Bewertungsbogen

Grundlegendes technisches Verständnis?

Java Basis-Knowhow

- Eigenschaften von speziellen Klassen
  - Object.equals() & Object.hashcode()
  - Enum
- Laufzeiteigenschaften von Objekten
  - Immutability
  - Memory-Allokation
- Datenstrukturen
  - Arrays
  - ArrayList, etc.
- Grundlegende Library-Kenntnisse
  - String-Operationen
  - Performance-Überlegungen
- Grundlegende Funktionen
  z.B. Collection-Interface
- Programmstruktur
  - Formattierung
  - Kommentare
  - Benennung

OO-Knowhow

- Kapselung
  Wie betreibt man Information-Hiding

- Vererbung
  - Wo macht sie Sinn?
  - Wo ist sie eher hinderlich?

- High Cohesion vs Low Coupling
  - Wo sind diese beiden Prinzipien bevorzugt anzuwenden?

Design/Architektur-Knowhow

- Design-Patterns
  - Singleton (gut / schlecht?)
  - Strategy Pattern
  - Welche Patterns kennt er?

- "Anämisches Domänen-Modell"
  - Gut / schlecht?
  - Wann tolerierbar?

- Nebenläufigkeit
  - Objekt-Zustände (was wäre wenn?)
  - Synchronisation (und ihre Auswirkungen)

- Schnittstellen
  - Prinzipien, die zu beachten sind
  - Was ist bei API-Änderung zu beachten

Testing-Knowhow

- Testvorgehen
  Zuerst Verifikation, dass Tests laufen, Tests als Doku verwendet?, Red - Green - Refactor

- Border Cases
  Sieht der Entwickler mehr?

- Was ist ein guter Test?
  Naming, arrange/act/assert, ...

- JUnit-Grundlagen
  Wie funktioniert die Library?

Agile-Knowhow

- Refactoring
  Definition, Unterschied zu Redesign?

- TDD
  - Wozu macht man das?
  - Aktuelle Kontroverse?

- Pair-Programming
  - Wozu soll das gut sein?
  - Seine Einstellung?

- Immer lauffähiger Code
  - Wie kann das erreicht werden?
  - Ist das überhaupt sinnvoll?

- Methoden
  - Unterschiede Scrum, XP, Kanban
  - Können sie sich ergänzen?

Arbeitstechnik

Sich einen Überblick verschaffen

- Was tut die Applikation?
  - Funktion erraten nach Code-Sichtung
- Klassen sichten
  - Domänenmodell
  - Logikteil
  - Tests
- Methoden in Logikteil
  - Unbenutzte Methoden
  - Duplikate
- Mikroplanung
  Taktische Schritte festgehalten bevor mit Coding begonnen (niedergeschrieben, im Kopf oder planlos)?
- Visuelle Unterstützung?
  Klassenmodell skizziert

Inkrementell arbeiten

- Baustellen
  Nur eine Änderung in Arbeit zu jedem Zeitpunkt
- Tests
  Laufen nach jedem Change?
- Änderung vollständig abgeschlossen
  Bleiben Restarbeiten übrig?
- Grösse der Änderung
  Wie gross ist eine Änderung, die am Stück gemacht wird?

Fokussierung

- Ablenkung
  - Lenkt jedes Gespräch ab?
  - Wie schnell kommt er wieder in den "Fluss"?
- Hilfsmittel
  - Werden Hilfsmittel genutzt?
  - z.B. Pomodoro-Technik

Autovalidierung

- Selbstkritik
  Stellt man sich ab und zu selbst in Frage?

- Abwägen von Entscheidungen
  Werde Alternativen systematisch abgewogen?

Qualitätsbewusstsein

- Technische Schuld
  Umgang damit

- Tradeoffs
  Informierte Design-Entscheide

- Over-Engineering
  Generische Lösung in der Annahme, dass es weitere Anwendungsfälle geben könnte

Tool-Bedienung

- Keyboard-Shortcuts
  - Welche kennt er?
  - Wie intensiv gebraucht er sie?

- Funktionen der IDE
  - Compare zweier Methoden (Diff)
  - Refactoring-Features

Sozialkompetenz

Aufgeschlossenheit/Kritikfähigkeit

- Reaktion auf Rat
  - Nimmt er passende Ratschläge dankbar an?
  - Würdigt er andere Meinungen gebührend?
  - Durchsetzen eigener Meinung?

- Meinung einholen
  - Holt er andere Meinungen ein, wenn er unsicher ist?

Entscheidungsfreudigkeit

- Unsicherheit
  - Paralyse/Blockade
  - Panik/Planlosigkeit
  - Kontrollierte Experimente (Exploration)
  - Rückfragen

- Kontolle
  - Backup-Strategien
  - Risikominimierungsmassnahmen

Neugierde

- Interesse
  - Stellt er Fragen, wie Dinge hier gelöst werden?
  - Stellt er Fragen, die er einem Manager nicht stellen kann?
  - Interessiert er sich für die Arbeitsumgebung, Tooling und Zusammenarbeit?
  - Legt er Wert auf agiles Vorgehen?

Weitere Beobachtungen

Kommunikation

- Deutsch
  Sprechen: verständlich, genau
- Englisch
  Lesen: verstehen
- Erklären
  - In der Lage Sachverhalt klar, deutlich zu erklären
  - Holt er Feedback der Zuhörer ein?
- Kommunikationsfreude
  Frage er nach, wenn er etwas nicht versteht?