|  |  |  |
| --- | --- | --- |
| **Lieferung 1 (24.02)** | **Keep**   * **dailies** | **Try**   * **Side-by-Side/ Pair Programming** * **Gemeinsame Iterationsplanung/ Abstimmung** * **Review** * **Testing** * **Datenbank gleich pflegen (lokal)** * **Häufigeres pushen** |
| **Problems**   * **mehr Kommunikation** * **lokale Datenbank** |
| **Lieferung 2 (10.03)** | **Keep**   * **dailies** * **Pair-Programming** * **Tests (JUnit)** | **Try**   * **Anwender zwischen durch fragen** * **PP ausbauen** * **Tickets für Probleme mit hoher Priorität** |
| **Problems**   * **Tests (JSFUnit)** * **Probleme zwar erkannt ABER nicht behoben** |
| **Lieferung 3 (24.03)** | **Keep**   * **dailies** * **Reviews** * **Tests (JUnit)** | **Try**   * **Git integrieren**   **🡪 Commits an Tickets hängen**  **🡪 leichterer Review** |
| **Problems**   * **Zu wenig Pair Programming 🡪 Zu wenig Zeit**   **🡪 Tasks zu klein/ leicht** |
| **Lieferung 4 (07.04)** | **Keep**   * **Bugs schnell in Tools augenommen** * **Schnelle Bearbeitung** * **Gute Aufgabenverteilung** * **Git integration** | **Try (für nächste Projekte)**   * **Anwender einbeziehen** * **PP:**   + **Korrekt durchführen**   + **In Blitzplanung für Tasks festlegen & mit abschätzen** * **Lieferungsplanung: kompletten Projektplan überarbeiten/berücksichtigen** * **Mit bekannten Techniken starten** * **Gitintegration Pflicht! 🡪 ausbaufähig (Zeit loggen,…)** * **Häufiges Pushen** * **Test explizit planen VON ANFANG AN** * **Automatic build (Tests, Metriken)** |
| **Problems**   * **PP 🡪 richtige Rollenverteilung** * **Anwender einbeziehen** * **Entwickler rät wie es sein soll 🡪 NACHFRAGEN!!!** * **Taskzuweisung** |