# Themensammlung

1. **Cloud**: Der Hohrer versteht was eine Cloud ist und wie sie funktionieren. IaaS, PaaS, SaaS und FaaS sollten dem Hoehrer nach der Presentation keine Fremdwoerter mehr sein Der Hoeher versteht welche Hosting Modelle es gibt und kennt die jeweiligen Vor- und Nachteile. (ca. 30min)
2. **Code Smells & Refactoring**: Der Hoeher weiss was Code Smells sind und wie man vorgeht beim Refaktorisieren. Der Hoeher lernt einige Code Smells und Refactorings kennen am Beispiel.
3. **Container:** Der Hoeher versteht den Unterschied zwischen Virtual Machine und Container. Der Hoherer lernt Docker kennen, eine Platform zur Containerisierung. Weiters versteht der Hoeher was eine Container-Orchestrierung ist und kennt damit Kubernetes. Die Einsatzgebiete und die Vorteile von Container sind dem Hoheher klar. *Dev-Containers werden speziell hervorgehoben.*
4. **Metriken:** Der Hohrer versteht was Software-Metriken sind und kennt ihr Einsatzgebiet. Der Nutzen ist fuer den Hoherer klar ersichtlich.
5. **Unit Testing 1:** Der Hoehrer kennt die Grundlagen von Testing und kann das Kosten-Nutzen-Verhaeltniss ziehen. Unit Testing und die dazugehorigen Sub-Themen wie Frameworks, Stubs, Mock, Fake sind klar. Der Hoehrer versteht was Code Coverage ist und kennt dessen Methoden. Weiters kennt der Hoehrer die Details zu Testcase Design & Management.
6. **Unit Testing 2:** Der Hoehrer weiss was Mockobjekte sind und weiss wann und wie er sie verwenden soll.
7. **Application Security:** Grundkonzepte der Applikation Security wie Symetrische Verschluesselung, Asymetrische Verschluesselung, Hashfunktion, Signatur sind dem Hoehrer klar. Er versteht auch Begriffe die zu Application Security zugehoerig sind. Auch weiss der Hoeherer wier er zur steigerung der Security beitragen kann.
8. **Design Patterns:** Der Hohrer kennt die Grundlagen von Design Patterns. Exemplarisch werden die GoF Patterns gelernt und der Hoehrer lern sie in einigen Beispielen anzuwenden.
9. **Clean Code Developer:** Der Hoeher hat einen Ueberblick ueber `clean coding`. Er kennt das Wertesystem, die Tugenden und die Grade.
10. **Git:** Der Hoehrer versteht was Git ist und wozu es gebraucht wird. Er kennt das Konzept und versteht die Begrifflichkeiten. Weiterhin kennt der Hoeher Basiskommandos. Branches, Branching-Strategien und Merge-Konflikte sind dem Hohrer keine Fremdwoerter mehr. EXTENDED: Der Hoehrer weiss was Tags sind und wie er sie einsetzten soll. Er kennt .gitignore, wie man Arbeit signed und versteht debugging mit git.
11. **Software Architekture:** Der Hoehrer versteht was Software Architektur ist und kennt die Aufgaben eines Software Architekten. Die Bedeutung von Software Architektur ist klar. Der Hoehrer kennt das moderene `Framework`, bzw. den Prozessablauf der Architektur-Arbeit.
12. **Vue.js**: Der Hoehrer lernt das web-frontend Framework und Oekosystem `Vue.js` kennen. Er lernt die Kernkonzepte von Vue (ractive data binding, component lifecycle, vue instance). Weiter Themen sind Template Syntax, Computed Properties and Watchers, Class and Style Bindings, Conditionals and List Rendering, Event Handling, Components
13. **Angular:** Der Hoehrer lernt das Web Framework Angular kennen. Dazu gehoerhen Components, Template Syntax, Directives, Data Binding, Dependency Injection, Signals, Routing. EXTENDED: State Management, Change Detection, Angular CLI, Progressive Web Apps
14. **From Code to Assembler:** Der Hohrer versteht was Assembler ist und wie aus Code Assembler generiert wird. Er versteht auch wozu man Assembler braucht und wie es zu Maschinencode umgewandelt wird. An einem Beispiel wird der Pfad vom Code bis zum Prozessor presentiert.