

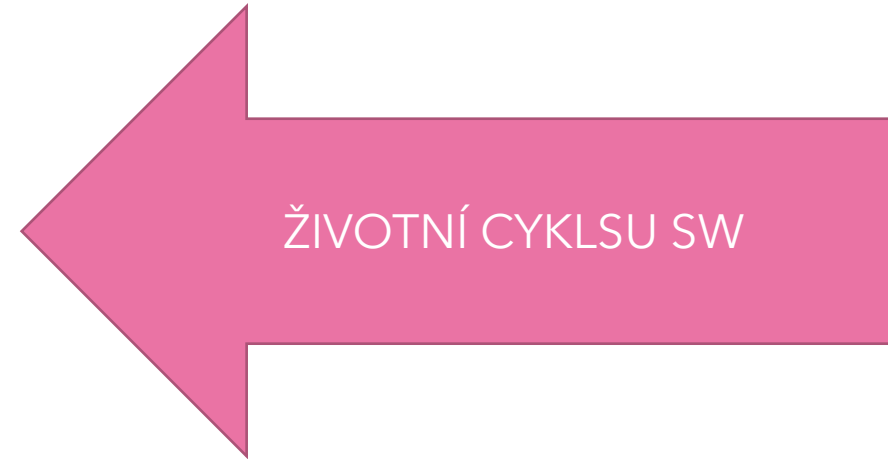
The background of the slide is a complex, abstract geometric pattern composed of numerous triangles of various sizes and colors, including shades of yellow, orange, red, pink, blue, and purple. The pattern is denser on the left side and fades into a dark grey/black background on the right. A small, solid pink horizontal bar is located in the upper right quadrant of the slide.

Vývoj her

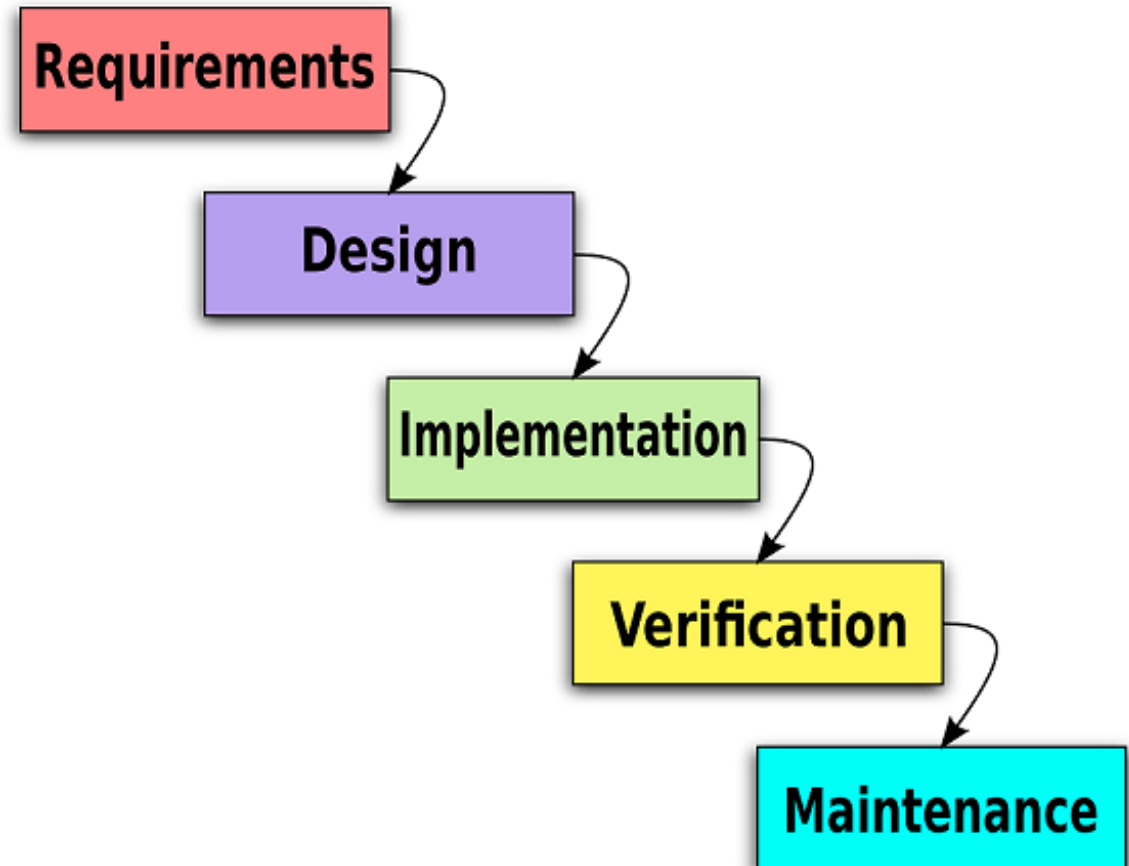
Z pohledu programátora

Vývoj software

- proces sestávající se z několika kroků:
 - Specifikace
 - Návrh
 - Implementace + dokumentace
 - Testování
 - Ladění
 - Údržba
- V závislosti na metodice vývoje na sebe jednotlivé kroky navazují případně probíhají do jisté míry paralelně
- **Agile** vs **Waterfall**



Agile vs Waterfall přístup vývoje



Nejpoužívanější jazyky pro vývoj her

- **C#** - díky síle Microsoftu silná podpora (X Box)
- **C++** - silný díky možnosti správě HW, zejména paměti (PS)
- **Java** - díky Java Virtual Machine (JVM) umožňuje vývoj pro velké množství platform
- **JavaScript** - vhodný zejména pro online a mobilní hry, snadná kombinace s HTML a CSS

Herní vývoj

- Vývoj her se podobá vývoji klasickým SW produktům
- Oproti klasickým SW má navíc audio, vizuální a herní složku
- Část vývoje věnující se samotné implementaci bývá přehlížena – překročení plánovaného rozpočtu
- Náročnost implementace závisí na množství již dříve vytvořeného - **engines**

Vývojový proces her

- Vývoj lze rozdělit zhruba do tří velkých částí
 - **Pre-produkce** - neprogramovací část
 - **Produkce**
 - První hratelná verze
 - Alfa verze
 - Code freeze
 - Beta
 - **Údržba**
 - Patching
 - Re-Release

Pre-produkce

- Začínáme prakticky na čistém papíře
- Sestavování konceptu hry
- Využití herních metodik
- Sestavování příběhu apod.
- Plánování a odhad času a financí nutných pro vývoj
- Cílem je vytvořit co nejpodrobnější popis a specifikace

Produkce

- Tato část je shodná s vývojem jakéhokoliv jiného SW
- Dodržujeme životní cyklus SW
- Mimo implementaci se zaměřujeme i na další části
 - Zvuk, design, dialogy, tvorba modelů, ...
- Výstupem této části je zdrojový kód a samotná hra v určité podobě

Produkce fáze testování

- **První hratelná verze** – hra z velké části obsahuje hlavní koncept
- **Alfa verze** – verze obsahuje všechny hlavní části, zde je prostor pro reimplementaci některých částí
- **Code freeze** – úprava pouze chyb
- **Beta verze** – hra je již téměř hotova a připravena na fázi vydání, opravují se zde pouze chyby
- Jakmile je hra připravena k vydání vytváří se tzv. **Gold Master**

Herní engine

- Framework, knihovna obsahující již implementované funkce potřebné pro herní vývoj
- Hotový engine často obsahuje spoustu modulů připravených pro použití
 - Rendering 2D/3D, detekce kolizí, fyzikální modely, zvuk, AI, práce s pamětí, ...
- Použití již hotového engine výrazně snižuje rozpočet

Příklad využití engine



Unity engine



- Jeden z nejpopulárnějších herních engine
- Zpřístupnění vývoje her pro každého
- Původně určeno pro vývoj na MacOS
- Engine využívající jak herní studia, tak nadšenci
- Možnost vyvíjet v různých programovacích jazycích
- Možnost vyvíjet pro různé platformy – PC, konzole, mobil, web, VR, ...

Příklady her vyvinutých s Unity engine

- Hearthstone
- Pokémon Go
- Cuphead
- Fallout Shelter
- League of Legends: Wild Rift
- Mario Kart Tour
- Cooking Simulator

