

# Softvér autopilota\*

Samuel Švec

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií

`xsvecs@stuba.sk`

25.10.2021

## Abstrakt

Tento článok bude o tom ako funguje autopilot v lietadlách či v lodiach a taktiež pokrok v automobilovom priemysle z hľadiska softvérovej stránky. Rozoberie aj aktuálne chyby v softvéroch, ktoré môžu ovplyvniť spoľahlivosť a bezpečnosť autopilota a tým aj zastaviť používanie daného softvéru.

## 1 Úvod

Autopilot nie je v dnešnej dobe nový pojem. S autopilotom sa vieme stretnúť aj v našom bežnom živote, a to napríklad v leteckej či vodnej doprave. V oblasti automobilového priemyslu však nie je dostatočne autopilot využívaný. V súčasnej dobe niekoľko popredných firiem pracuje na vytvorení automobilového autopilota, ako napríklad Tesla. Avšak, použitie autopilota prináša aj veľa hrozieb. Základný problém, ktorý bol naznačený v úvode, je podrobnejšie vysvetlený v časti 3. Záverečné poznámky sa nachádzajú v časti 6.

## 2 Ako funguje autopilot

Z obr. 1 je všetko jasné.

Aj text môže byť prezentovaný ako obrázok. Stane sa z neho označný plávajúci objekt. Po vytvorení diagramu zrušte znak % pred príkazom `\includegraphics` označte tento riadok ako komentár (tiež pomocou znaku %).

Obr. 1: Rozhodujúci argument.

---

\*Semestrálny projekt v predmete Metódy inžinierskej práce, ak. rok 2021/22, vedenie: Vladimír Mlynarovič

### 3 Chyby autopilota

Základným problémom je teda... Najprv sa pozrieme na nejaké vysvetlenie (časť 3.1), a potom na ešte nejaké (časť 3.1).<sup>1</sup>

Môže sa zdať, že problém vlastne nejestvuje [1], ale bolo dokázané, že to tak nie je [2,3]. Napriek tomu, aj dnes na webe narazíme na všelijaké pochybné názory [4]. Dôležité veci možno *zdôrazniť kurzívou*.

#### 3.1 Nejaké vysvetlenie

Niekedy treba uviesť zoznam:

- jedna vec
- druhá vec

– x

– y

Ten istý zoznam, len číslovaný:

1. jedna vec
2. druhá vec

(a) x

(b) y

#### 3.2 Ešte nejaké vysvetlenie

**Veľmi dôležitá poznámka.** Niekedy je potrebné nadpisom označiť odsek. Text pokračuje hneď za nadpisom.

### 4 Dôležitá časť

### 5 Ešte dôležitejšia časť

### 6 Záver

## Literatúra

- [1] James O. Coplien. *Multi-Paradigm Design for C++*. Addison-Wesley, 1999.
- [2] Krzysztof Czarnecki, Simon Helsen, and Ulrich Eisenecker. Staged configuration through specialization and multi-level configuration of feature models. *Software Process: Improvement and Practice*, 10:143–169, April/June 2005.
- [3] Krzysztof Czarnecki and Chang Hwan Peter Kim. Cardinality-based feature modeling and constraints: A progress report. In *International Workshop on Software Factories, OOPSLA 2005*, San Diego, USA, October 2005.

---

<sup>1</sup>Niekedy môžete potrebovať aj poznámku pod čiarou.

- [4] Carnegie Mellon University Software Engineering Institute. A framework for software product line practice—version 5.0. [http://www.sei.cmu.edu/productlines/frame\\_report/](http://www.sei.cmu.edu/productlines/frame_report/).