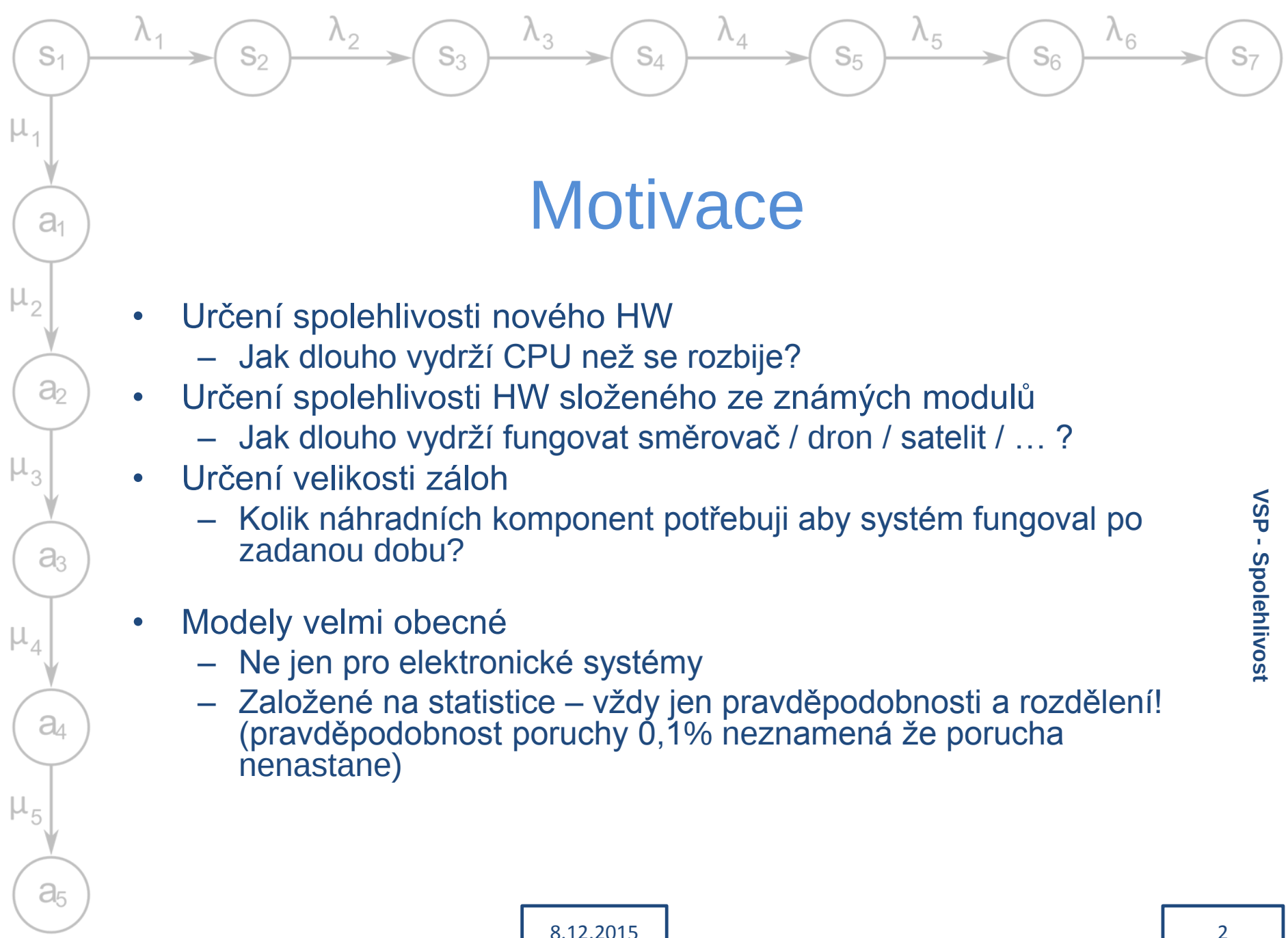
Modelování spolehlivost

Výkonnost a spolehlivost – KIV/VSP

Richard Lipka

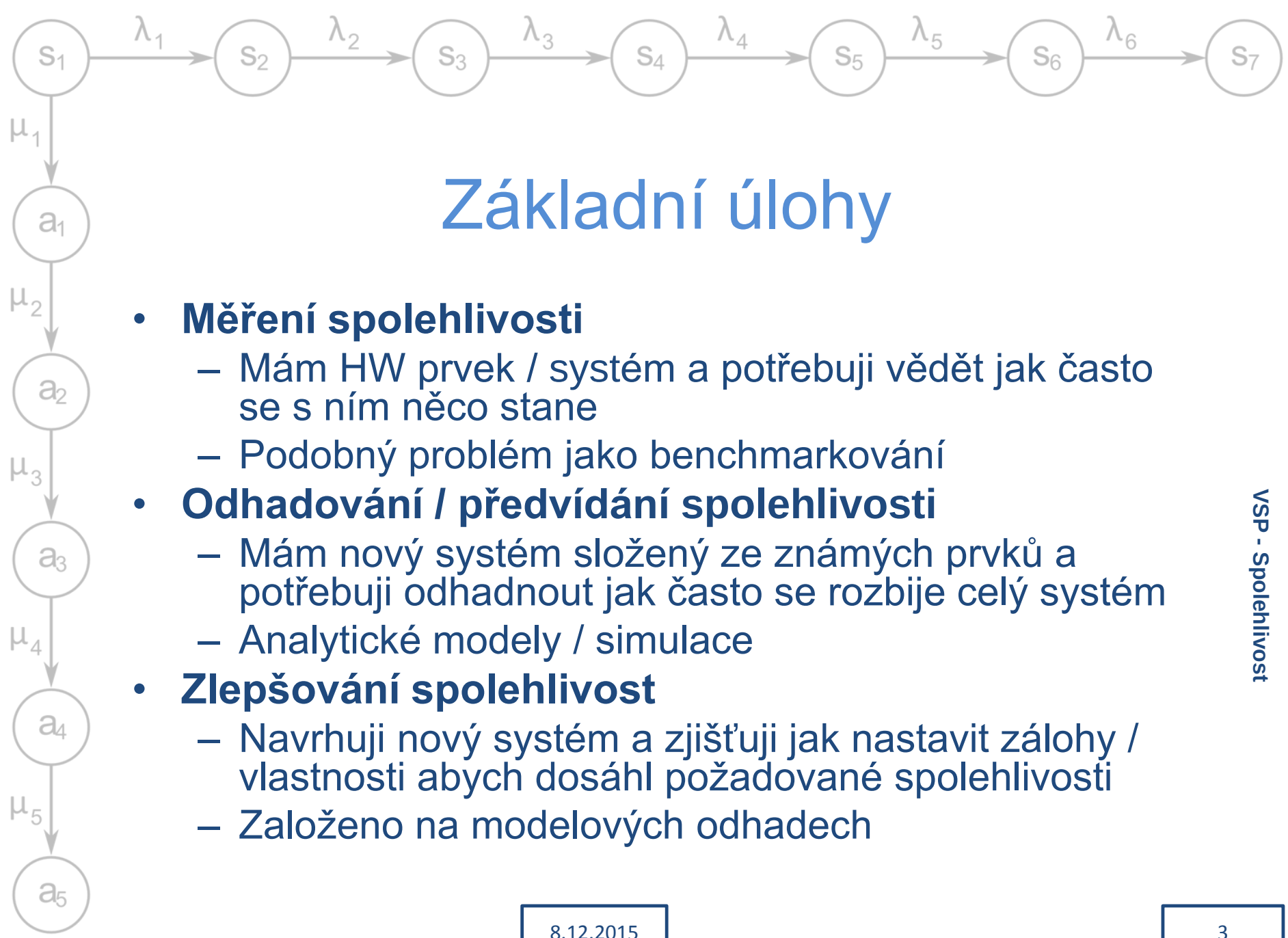
8.12.2015



Motivace

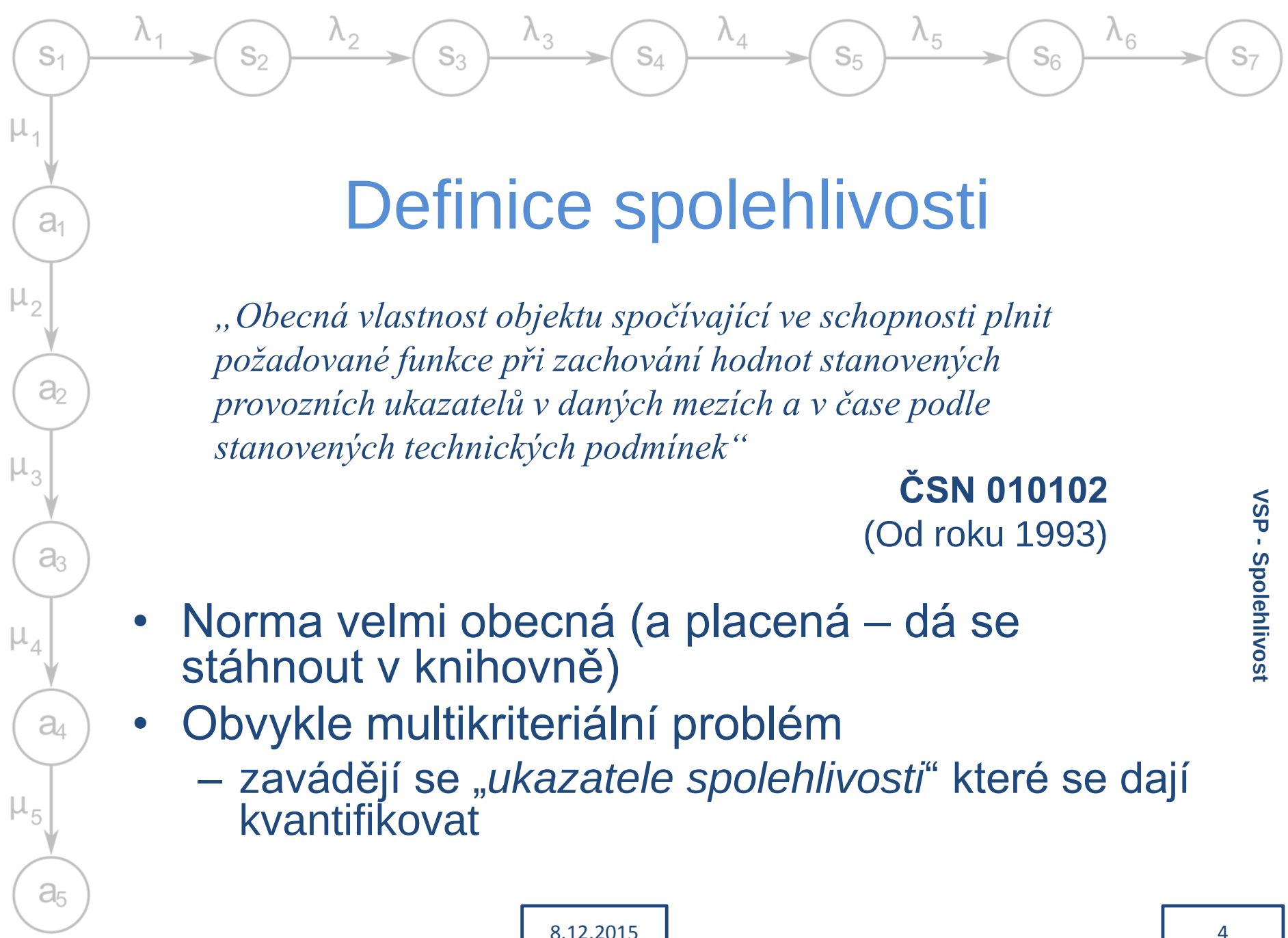
- Určení spolehlivosti nového HW
  - Jak dlouho vydrží CPU než se rozbije?
- Určení spolehlivosti HW složeného ze známých modulů
  - Jak dlouho vydrží fungovat směrovač / dron / satelit / … ?
- Určení velikosti záloh
  - Kolik náhradních komponent potřebuji aby systém fungoval po zadanou dobu?

- Modely velmi obecné
  - Ne jen pro elektronické systémy
  - Založené na statistice – vždy jen pravděpodobnosti a rozdělení! (pravděpodobnost poruchy 0,1% neznamená že porucha nenastane)



Základní úlohy

- **Měření spolehlivosti**
  - Mám HW prvek / systém a potřebuji vědět jak často se s ním něco stane
  - Podobný problém jako benchmarkování
- **Odhadování / předvídání spolehlivosti**
  - Mám nový systém složený ze známých prvků a potřebuji odhadnout jak často se rozbije celý systém
  - Analytické modely / simulace
- **Zlepšování spolehlivost**
  - Navrhuji nový systém a zjišťuji jak nastavit zálohy / vlastnosti abych dosáhl požadované spolehlivosti
  - Založeno na modelových odhadech

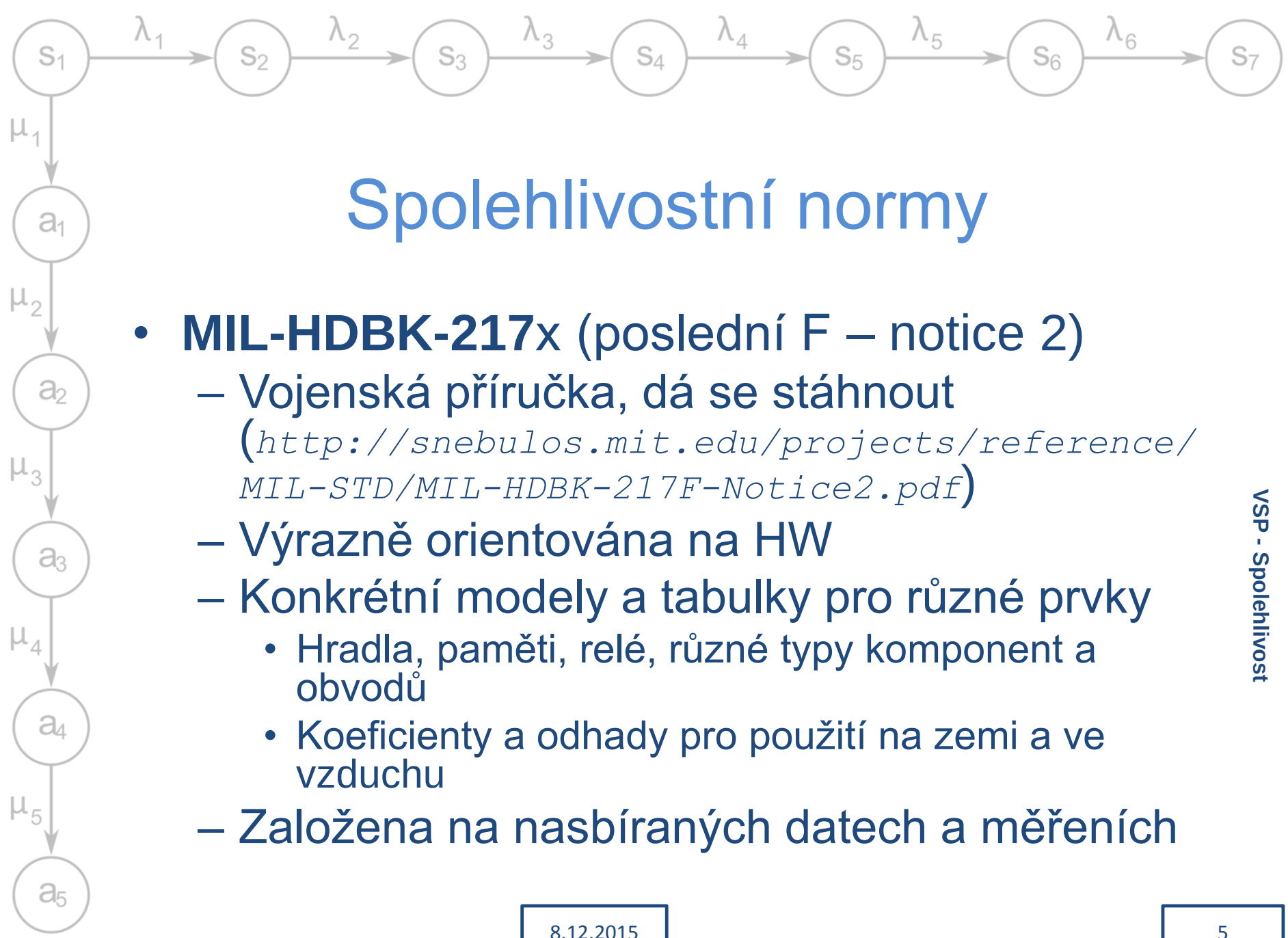


Definice spolehlivosti

„Obecná vlastnost objektu spočívající ve schopnosti plnit požadované funkce při zachování hodnot stanovených provozních ukazatelů v daných mezích a v čase podle stanovených technických podmínek“

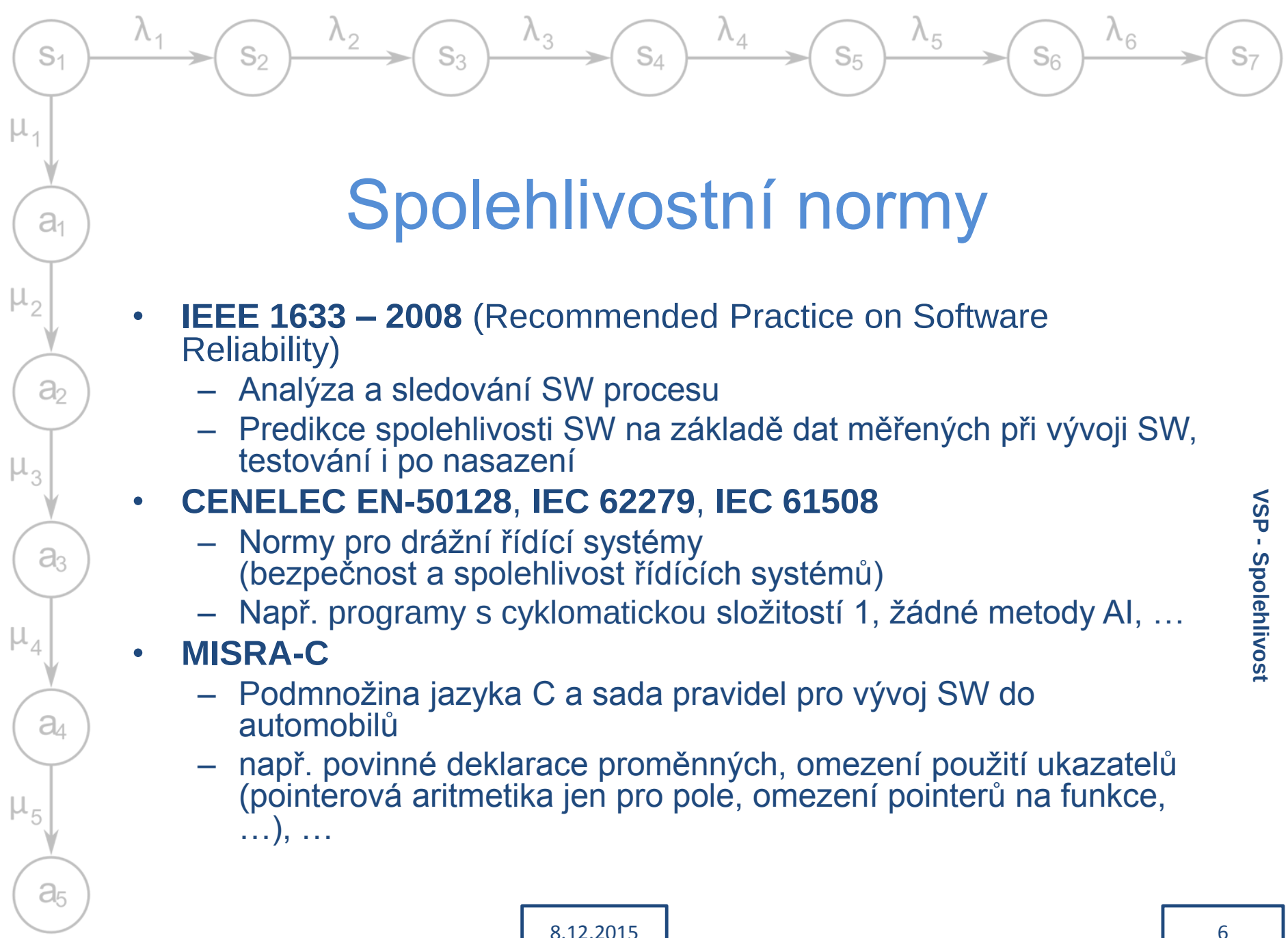
ČSN 010102
(Od roku 1993)

- Norma velmi obecná (a placená – dá se stáhnout v knihovně)
- Obvykle multikriteriální problém
  - zavádějí se „*ukazatele spolehlivosti*“ které se dají kvantifikovat



Spolehlivostní normy

- **MIL-HDBK-217**x (poslední F – notice 2)
  - Vojenská příručka, dá se stáhnout (*http://snebulos.mit.edu/projects/reference/MIL-STD/MIL-HDBK-217F-Notice2.pdf*)
  - Výrazně orientována na HW
  - Konkrétní modely a tabulky pro různé prvky
    - Hradla, paměti, relé, různé typy komponent a obvodů
    - Koeficienty a odhady pro použití na zemi a ve vzduchu
  - Založena na nasbíraných datech a měřeních



Spolehlivostní normy

- **IEEE 1633 – 2008** (Recommended Practice on Software Reliability)
  - Analýza a sledování SW procesu
  - Predikce spolehlivosti SW na základě dat měřených při vývoji SW, testování i po nasazení
- **CENELEC EN-50128**, **IEC 62279**, **IEC 61508**
  - Normy pro drážní řídící systémy (bezpečnost a spolehlivost řídících systémů)
  - Např. programy s cyklomatickou složitostí 1, žádné metody AI, …
- **MISRA-C**
  - Podmnožina jazyka C a sada pravidel pro vývoj SW do automobilů
  - např. povinné deklarace proměnných, omezení použití ukazatelů (pointerová aritmetika jen pro pole, omezení pointerů na funkce, …), …

Základní pojmy

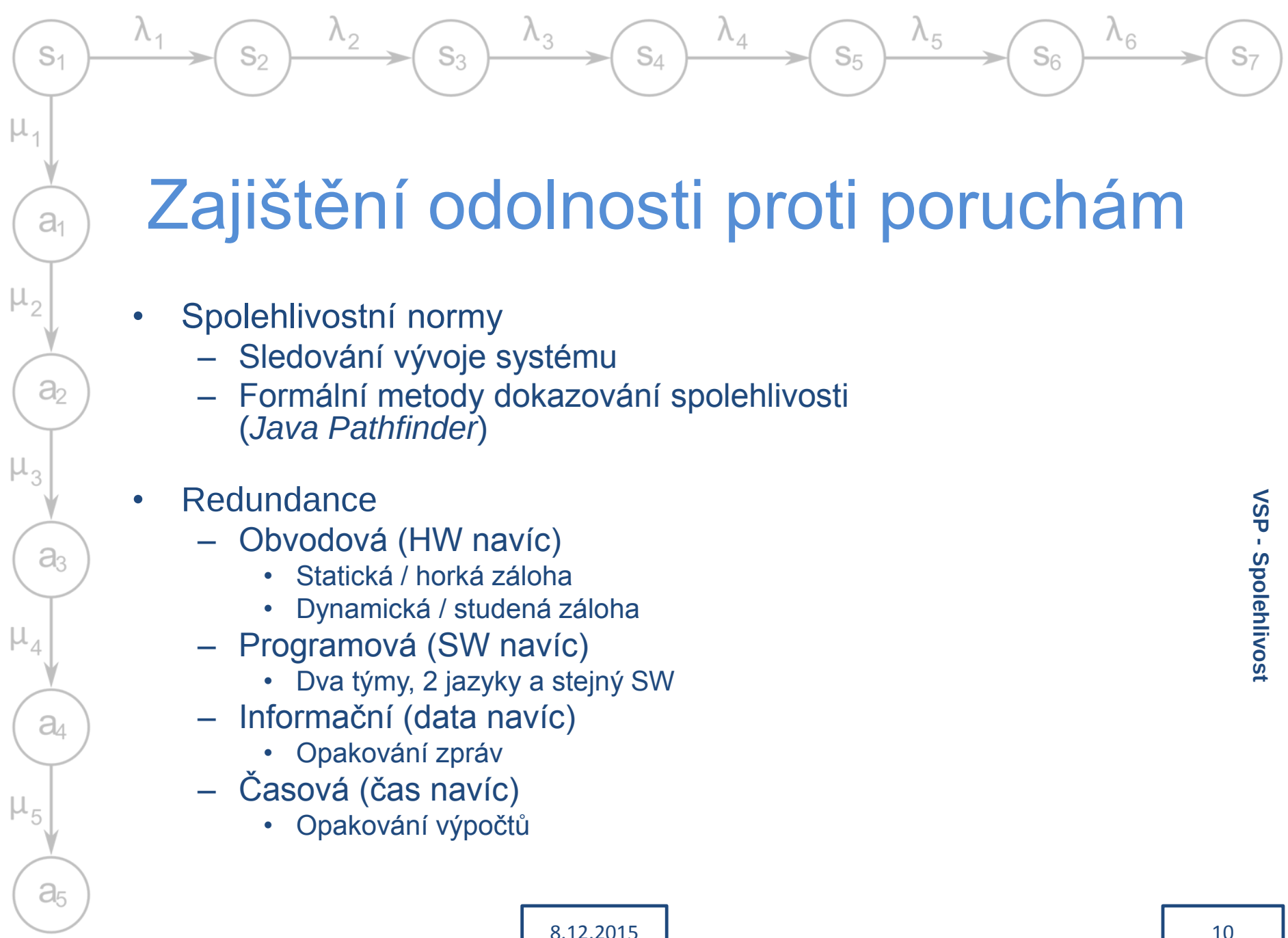
- **Poruchy**:
  - Systematické – chyby v návrhu systému (zapomenuté kontroly, chyby operátora, přehřívání, …)
  - Náhodné – vnější vlivy mimo kontrolu (nárazy, počasí, radiace, …)
- **Stavy**:
  - Bezporuchový – systém normálně funguje
  - Poruchový – systém neposkytuje požadovanou službu (dokážu ho detekovat?)
- **Systémy**:
  - Obnovované – mohu je nějak opravit
  - Neobnovované – po poruše jsou zničeny (neopravitelný HW, nedostupný systém, příliš nákladné, …)

Klasifikace chyb

- Vznik a propagace chyby (různé terminologie)
  - *Mistake* – chyba operátora
  - *Fault* – domnělá příčina chyby (vnější vlivy, chyba v návrhu systému, …)
    - *Bizantine fault* – nevede k detekci selhání ale ke špatnému výsledku
  - *Error* – rozdíl mezi správným a získaným výsledkem
  - *Failure* – selhání systému – nedodržení specifikace
- Podle trvání
  - Stálé (*permanent*) – systém přestane fungovat
  - Přechodné (*transient*) – jednorázové selhání (zpráva nedorazí)
  - Opakující se (*intermintent*) – opakovaně se objevují a mizí

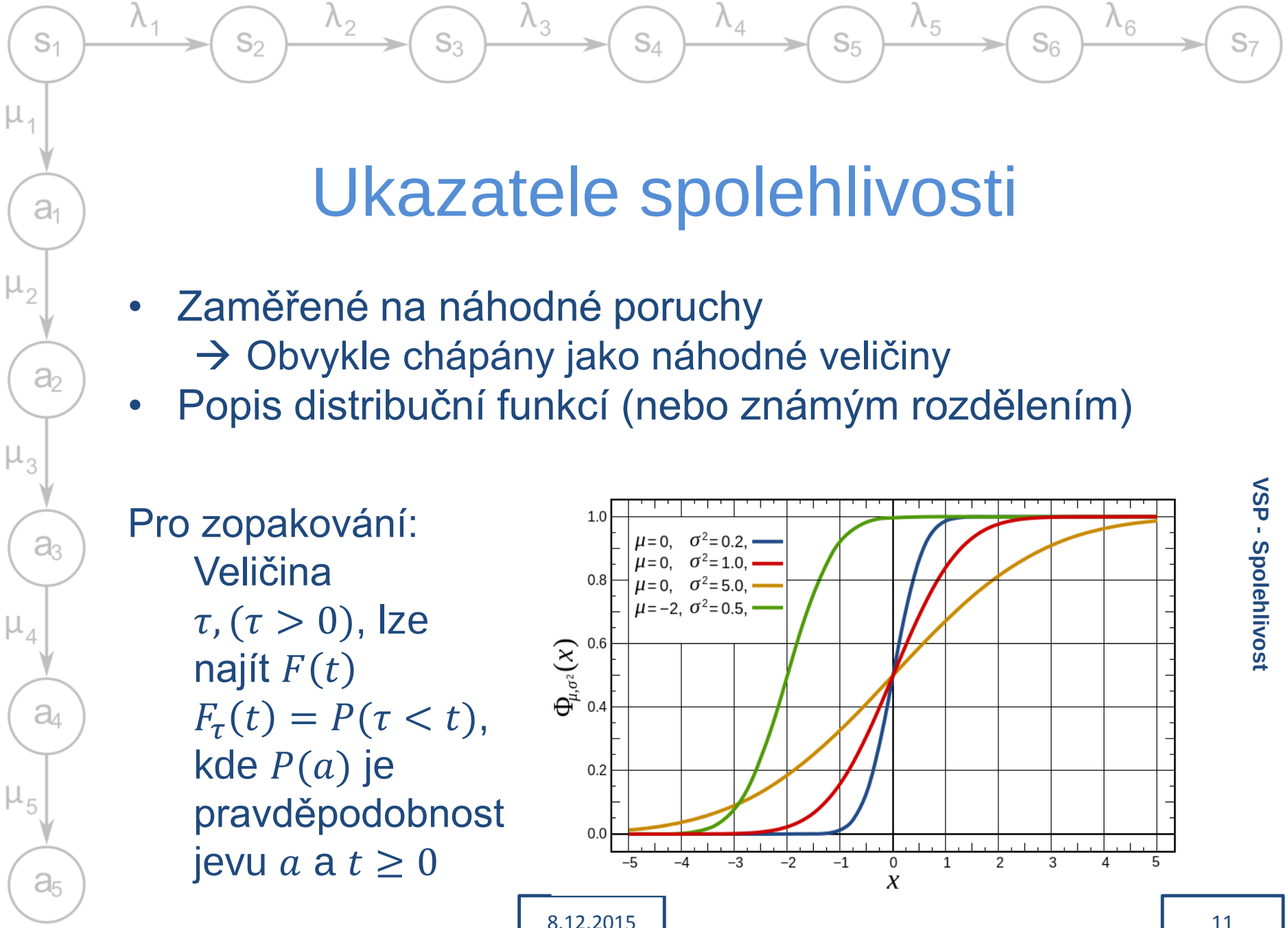
Kategorie spolehlivosti

- **Nezabezpečené systémy** – *fault avoidance*
  - „dělám vše co nejlépe“
- **Zabezpečené systémy** – *fail safe*
  - po poruše přepnutí do bezpečného stavu
  - Řízení křižovatky, drážní provoz, …
- **Systémy odolné proti poruchám** – *fail tolerant*
  - Neexistuje bezpečný stav → porucha „nesmí nastat“
  - Řídící systémy letadla, zabezpečení letového provozu



Zajištění odolnosti proti poruchám

- Spolehlivostní normy
  - Sledování vývoje systému
  - Formální metody dokazování spolehlivosti (*Java Pathfinder*)
- Redundance
  - Obvodová (HW navíc)
    - Statická / horká záloha
    - Dynamická / studená záloha
  - Programová (SW navíc)
    - Dva týmy, 2 jazyky a stejný SW
  - Informační (data navíc)
    - Opakování zpráv
  - Časová (čas navíc)
    - Opakování výpočtů

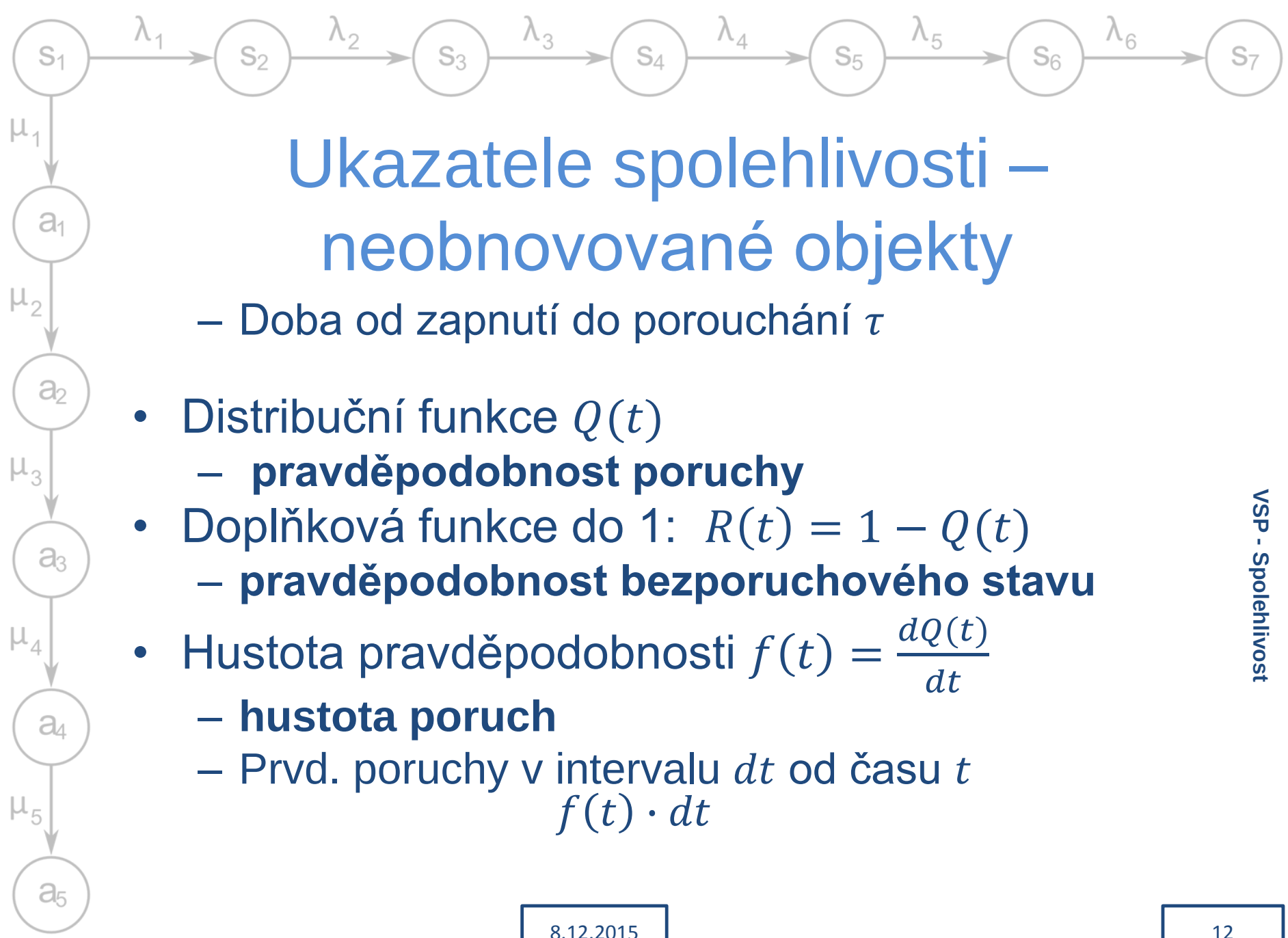


Ukazatele spolehlivosti

- Zaměřené na náhodné poruchy
  → Obvykle chápány jako náhodné veličiny
- Popis distribuční funkcí (nebo známým rozdělením)

Pro zopakování:

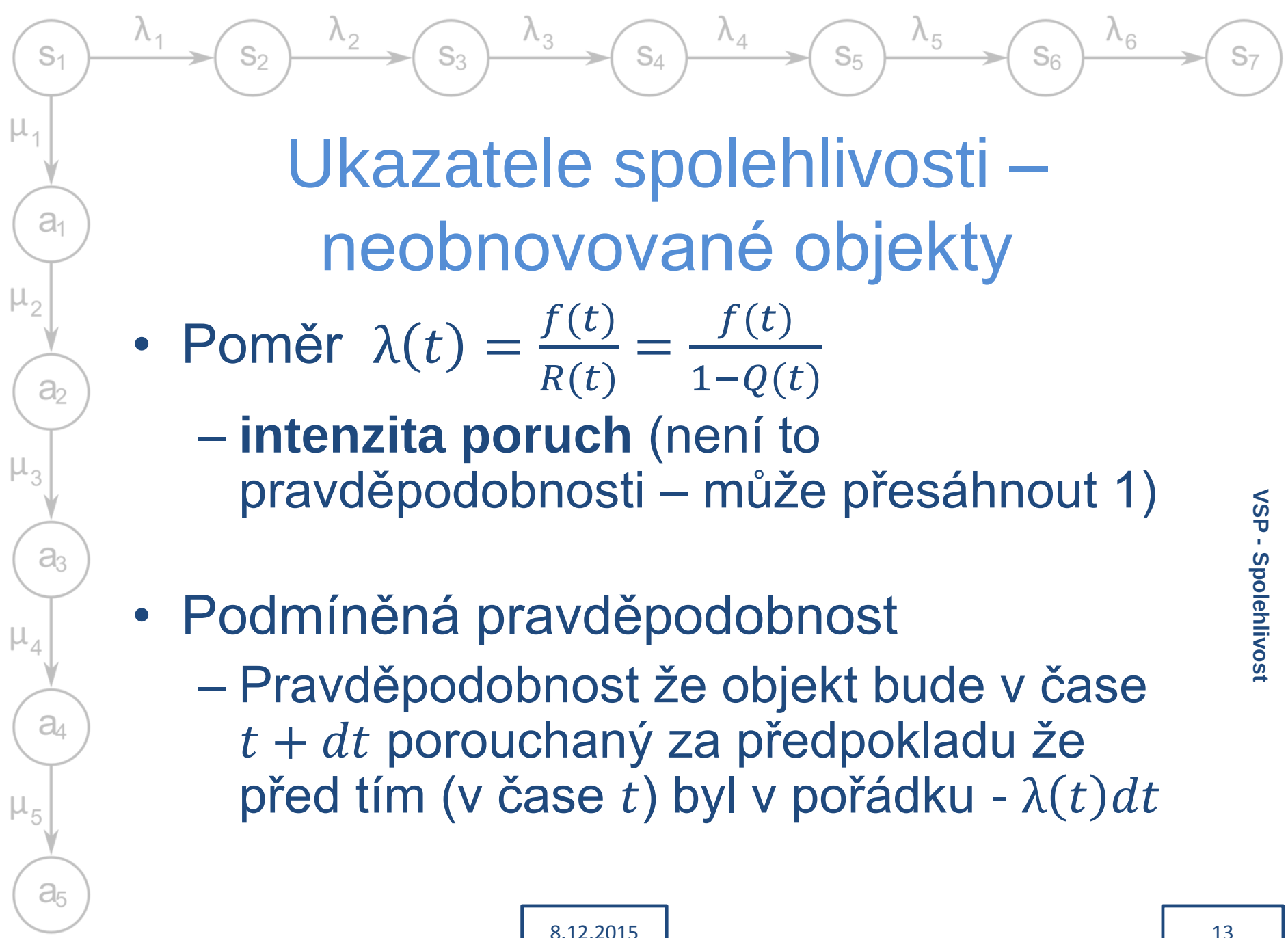
Veličina $\tau, (\tau > 0)$, lze najít $F(t)$ $F_\tau(t) = P(\tau < t)$, kde $P(a)$ je pravděpodobnost jevu a a $t \geq 0$



Ukazatele spolehlivosti – neobnovované objekty

– Doba od zapnutí do porouchání τ

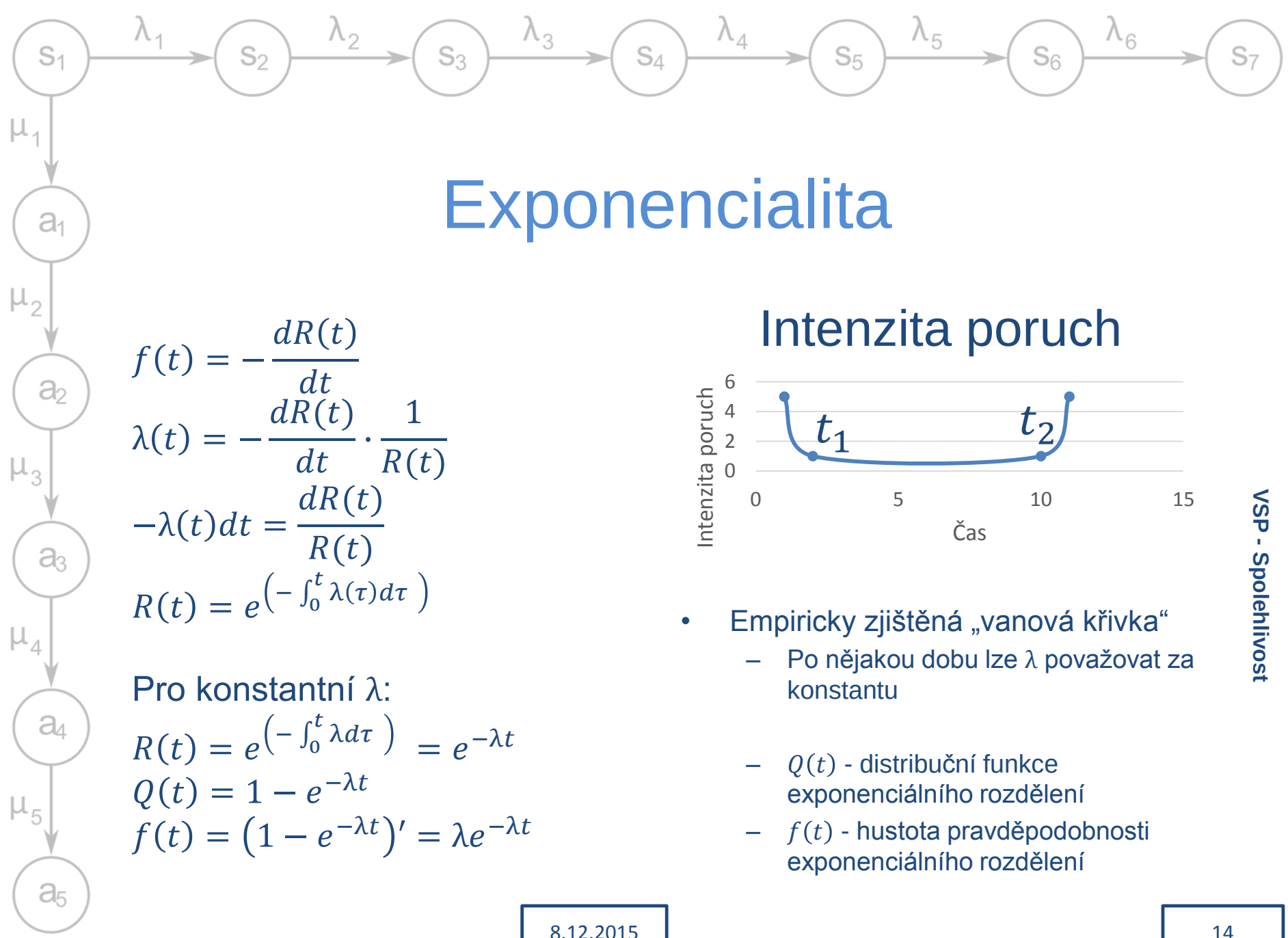
- Distribuční funkce $Q(t)$
  – **pravděpodobnost poruchy**
- Doplňková funkce do 1: $R(t) = 1 - Q(t)$
  – **pravděpodobnost bezporuchového stavu**
- Hustota pravděpodobnosti $f(t) = \frac{dQ(t)}{dt}$
  – **hustota poruch**
  – Prvd. poruchy v intervalu dt od času t
  $f(t) \cdot dt$



Ukazatele spolehlivosti – neobnovované objekty

- Poměr $\lambda(t) = \frac{f(t)}{R(t)} = \frac{f(t)}{1-Q(t)}$
  - **intenzita poruch** (není to pravděpodobnosti – může přesáhnout 1)

- Podmíněná pravděpodobnost
  - Pravděpodobnost že objekt bude v čase $t + dt$ porouchaný za předpokladu že před tím (v čase t) byl v pořádku - $\lambda(t)dt$



Exponencialita

$$f(t) = -\frac{dR(t)}{dt}$$

$$\lambda(t) = -\frac{dR(t)}{dt} \cdot \frac{1}{R(t)}$$

$$-\lambda(t)dt = \frac{dR(t)}{R(t)}$$

$$R(t) = e^{\left(-\int_0^t \lambda(\tau)d\tau\right)}$$

Pro konstantní λ:

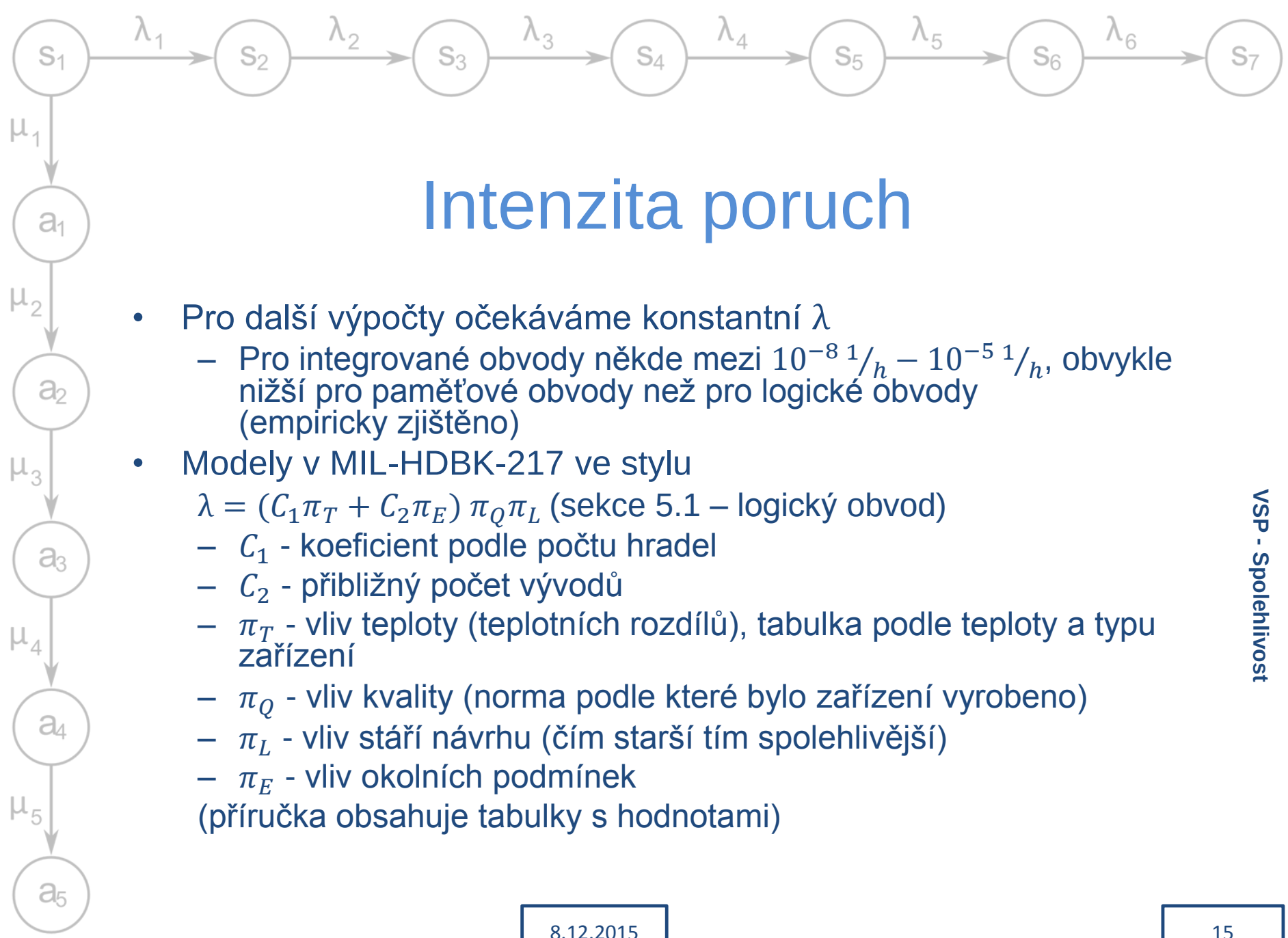
$$R(t) = e^{\left(-\int_0^t \lambda d\tau\right)} = e^{-\lambda t}$$

$$Q(t) = 1 - e^{-\lambda t}$$

$$f(t) = \left(1 - e^{-\lambda t}\right)' = \lambda e^{-\lambda t}$$

Intenzita poruch

- Empiricky zjištěná „vanová křivka“
  - Po nějakou dobu lze λ považovat za konstantu
  - $Q(t)$ - distribuční funkce exponenciálního rozdělení
  - $f(t)$ - hustota pravděpodobnosti exponenciálního rozdělení

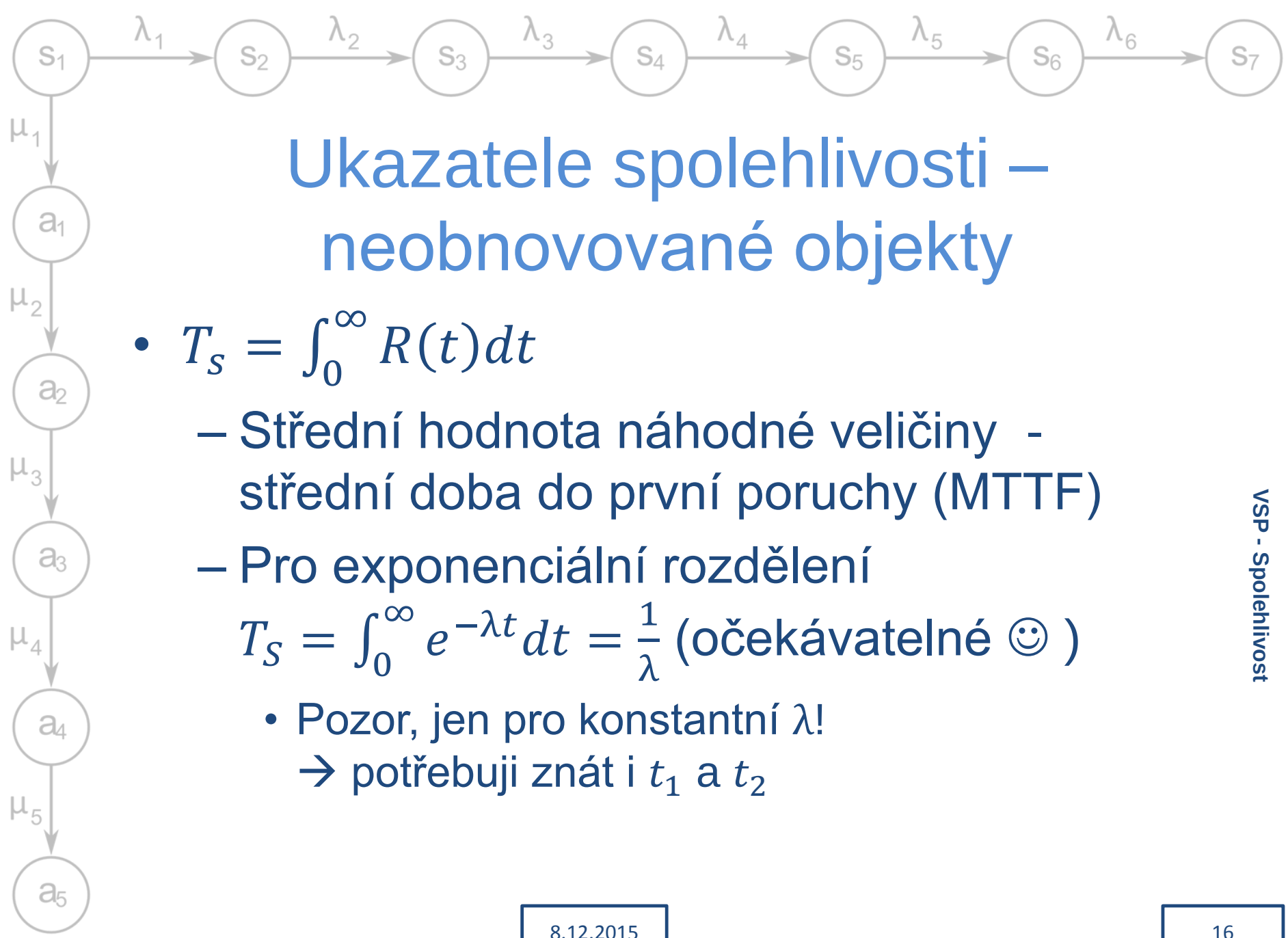


Intenzita poruch

- Pro další výpočty očekáváme konstantní λ
  - Pro integrované obvody někde mezi $10^{-8}\,{}^{1}/_{h} - 10^{-5}\,{}^{1}/_{h}$, obvykle nižší pro paměťové obvody než pro logické obvody (empiricky zjištěno)
- Modely v MIL-HDBK-217 ve stylu

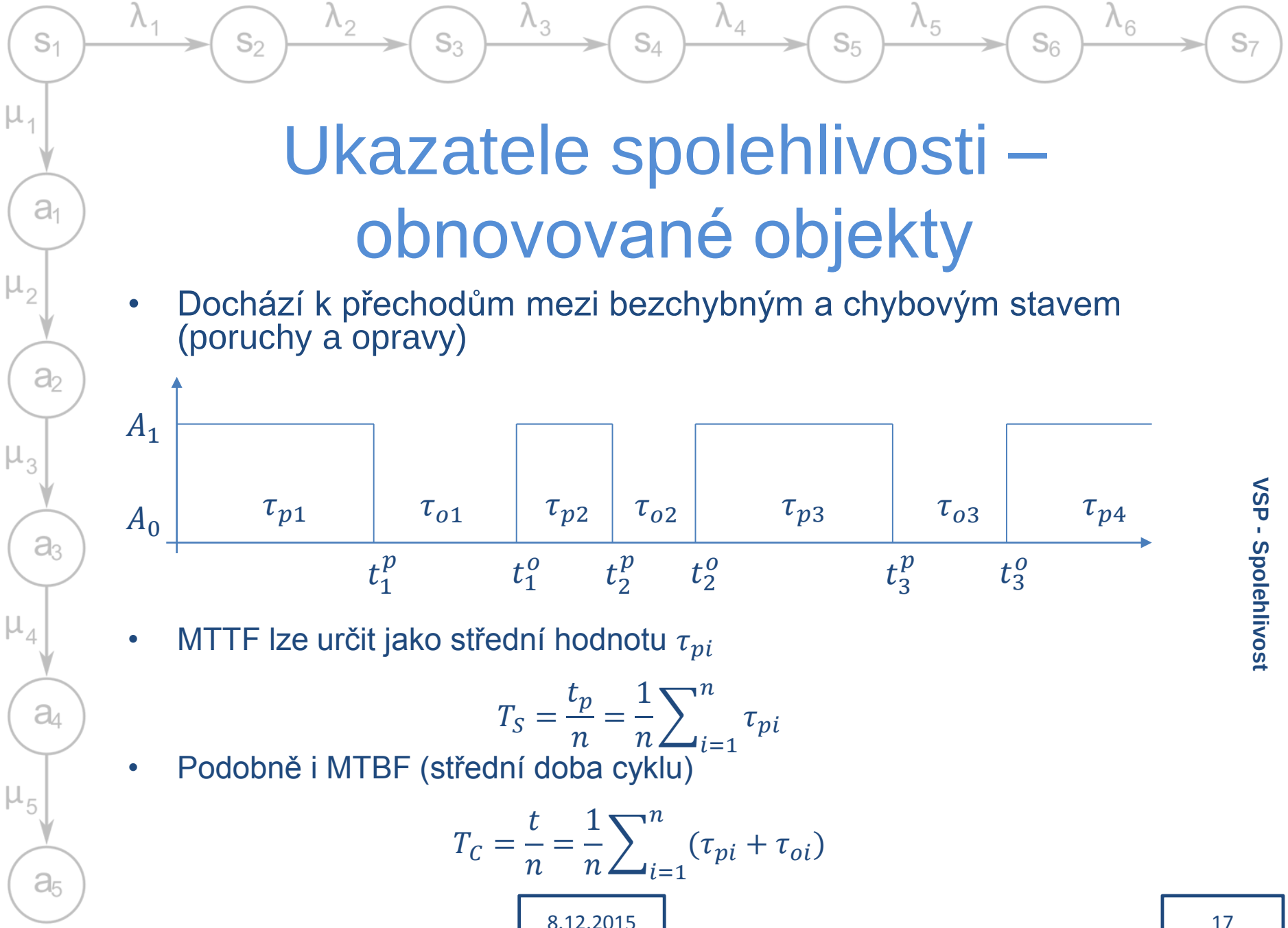
  $\lambda = (C_1\pi_T + C_2\pi_E)\,\pi_Q\pi_L$ (sekce 5.1 – logický obvod)
  - C_1 - koeficient podle počtu hradel
  - C_2 - přibližný počet vývodů
  - π_T - vliv teploty (teplotních rozdílů), tabulka podle teploty a typu zařízení
  - π_Q - vliv kvality (norma podle které bylo zařízení vyrobeno)
  - π_L - vliv stáří návrhu (čím starší tím spolehlivější)
  - π_E - vliv okolních podmínek

  (příručka obsahuje tabulky s hodnotami)



Ukazatele spolehlivosti – neobnovované objekty

- $T_s = \int_0^\infty R(t)dt$
  - Střední hodnota náhodné veličiny - střední doba do první poruchy (MTTF)
  - Pro exponenciální rozdělení
    $T_S = \int_0^\infty e^{-\lambda t}dt = \frac{1}{\lambda}$ (očekávatelné ☺)
    - Pozor, jen pro konstantní λ!
      → potřebuji znát i t_1 a t_2



Ukazatele spolehlivosti – obnovované objekty

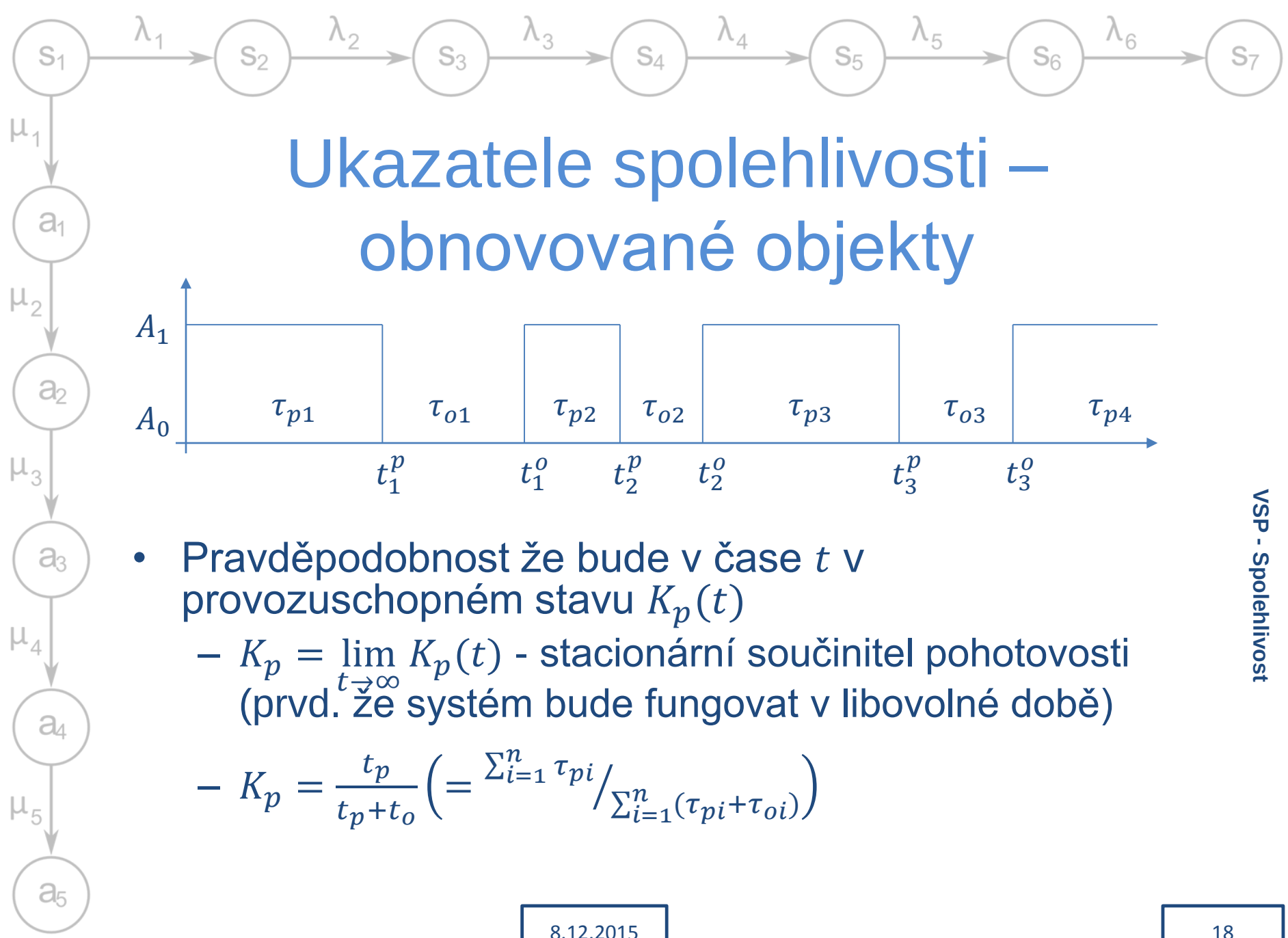
- Dochází k přechodům mezi bezchybným a chybovým stavem (poruchy a opravy)

- MTTF lze určit jako střední hodnotu τ_{pi}

$$T_S = \frac{t_p}{n} = \frac{1}{n}\sum_{i=1}^{n} \tau_{pi}$$

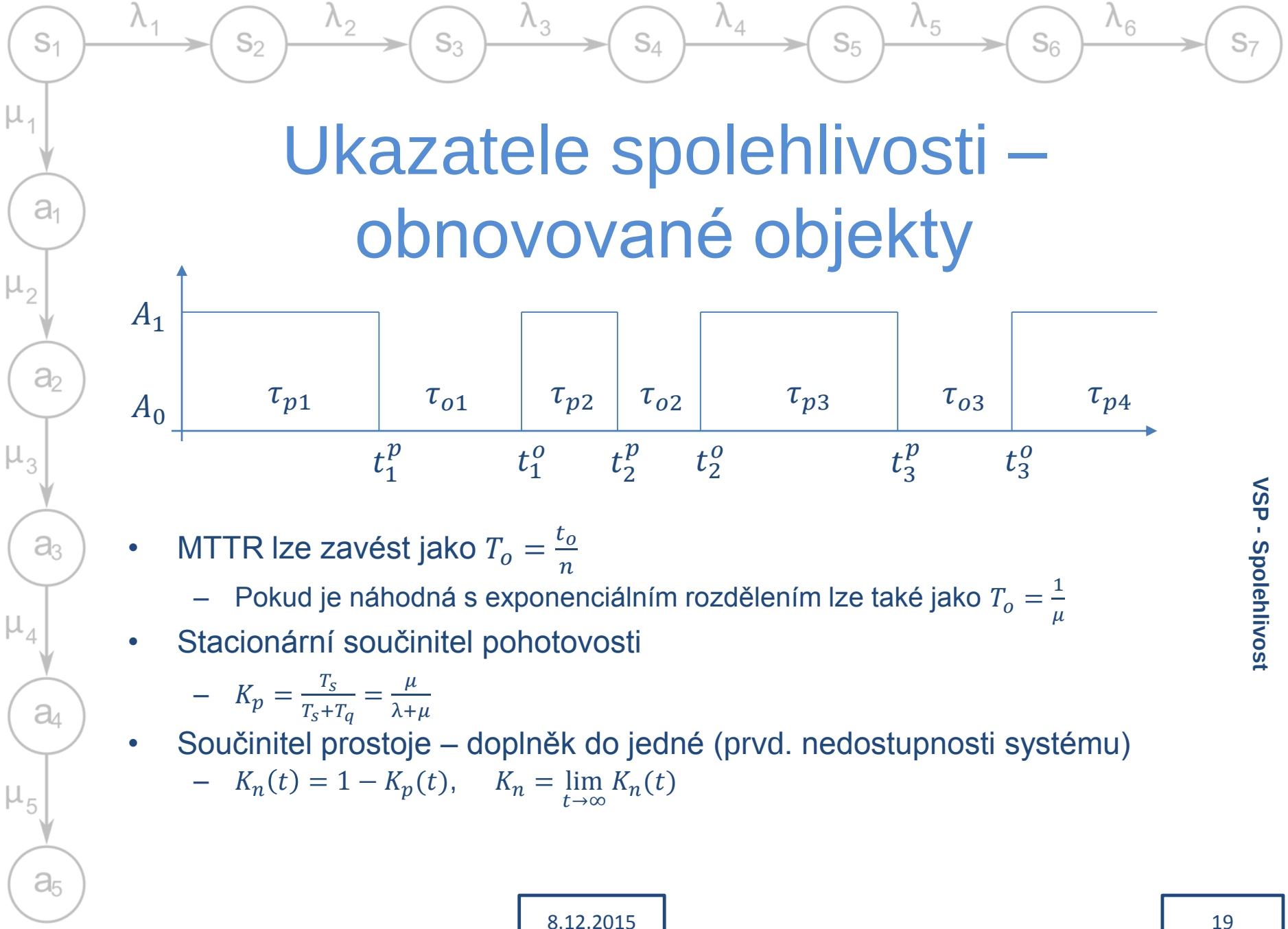
- Podobně i MTBF (střední doba cyklu)

$$T_C = \frac{t}{n} = \frac{1}{n}\sum_{i=1}^{n} (\tau_{pi} + \tau_{oi})$$



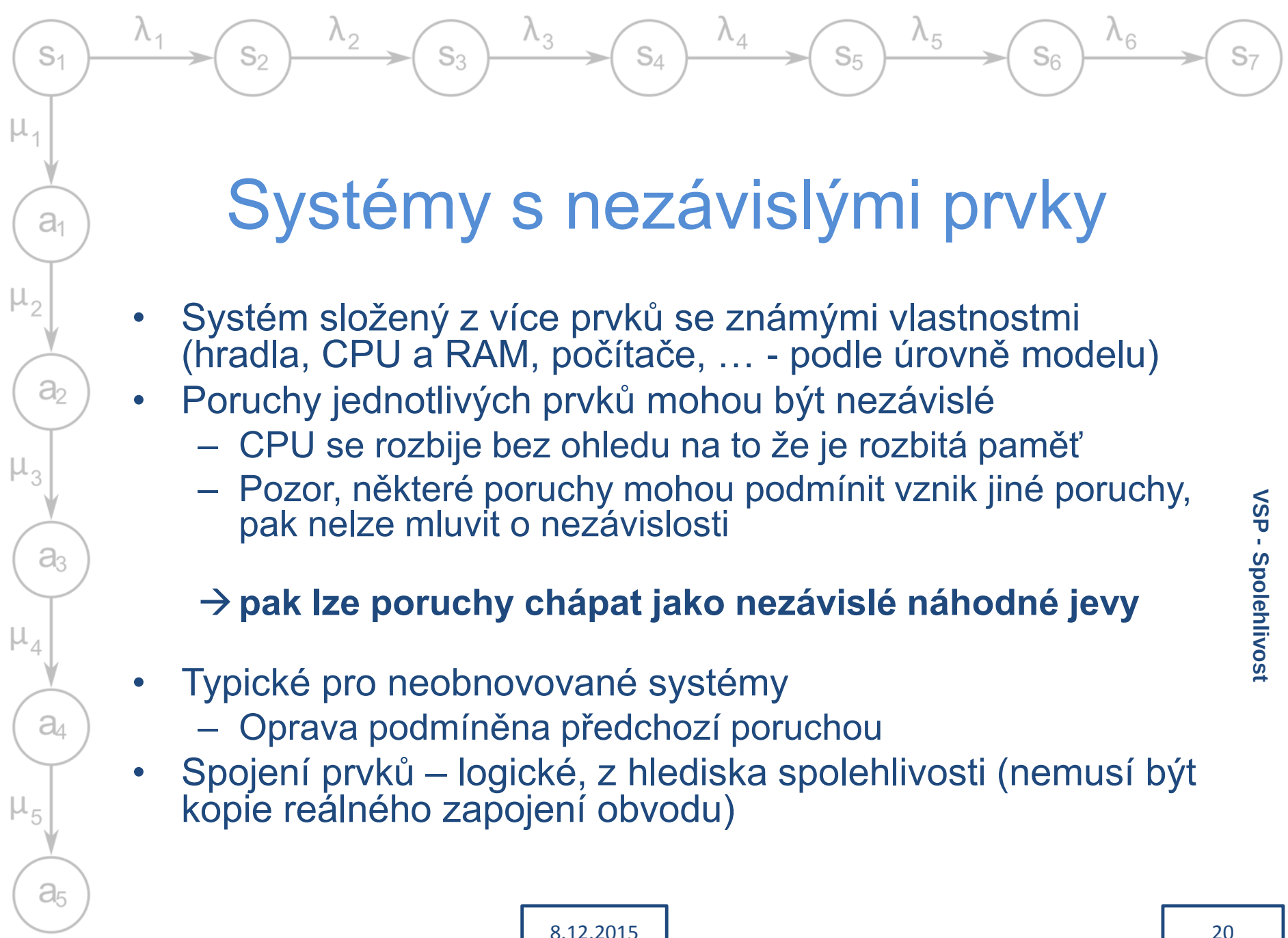
Ukazatele spolehlivosti – obnovované objekty

- Pravděpodobnost že bude v čase t v provozuschopném stavu $K_p(t)$
  - $K_p = \lim_{t \to \infty} K_p(t)$ - stacionární součinitel pohotovosti (prvd. že systém bude fungovat v libovolné době)
  - $K_p = \frac{t_p}{t_p + t_o} \left(= {\sum_{i=1}^{n} \tau_{pi}} / {\sum_{i=1}^{n} (\tau_{pi} + \tau_{oi})} \right)$



Ukazatele spolehlivosti – obnovované objekty

- MTTR lze zavést jako $T_o = \frac{t_o}{n}$
  - Pokud je náhodná s exponenciálním rozdělením lze také jako $T_o = \frac{1}{\mu}$
- Stacionární součinitel pohotovosti
  - $K_p = \frac{T_s}{T_s + T_q} = \frac{\mu}{\lambda + \mu}$
- Součinitel prostoje – doplněk do jedné (prvd. nedostupnosti systému)
  - $K_n(t) = 1 - K_p(t), \quad K_n = \lim_{t \to \infty} K_n(t)$

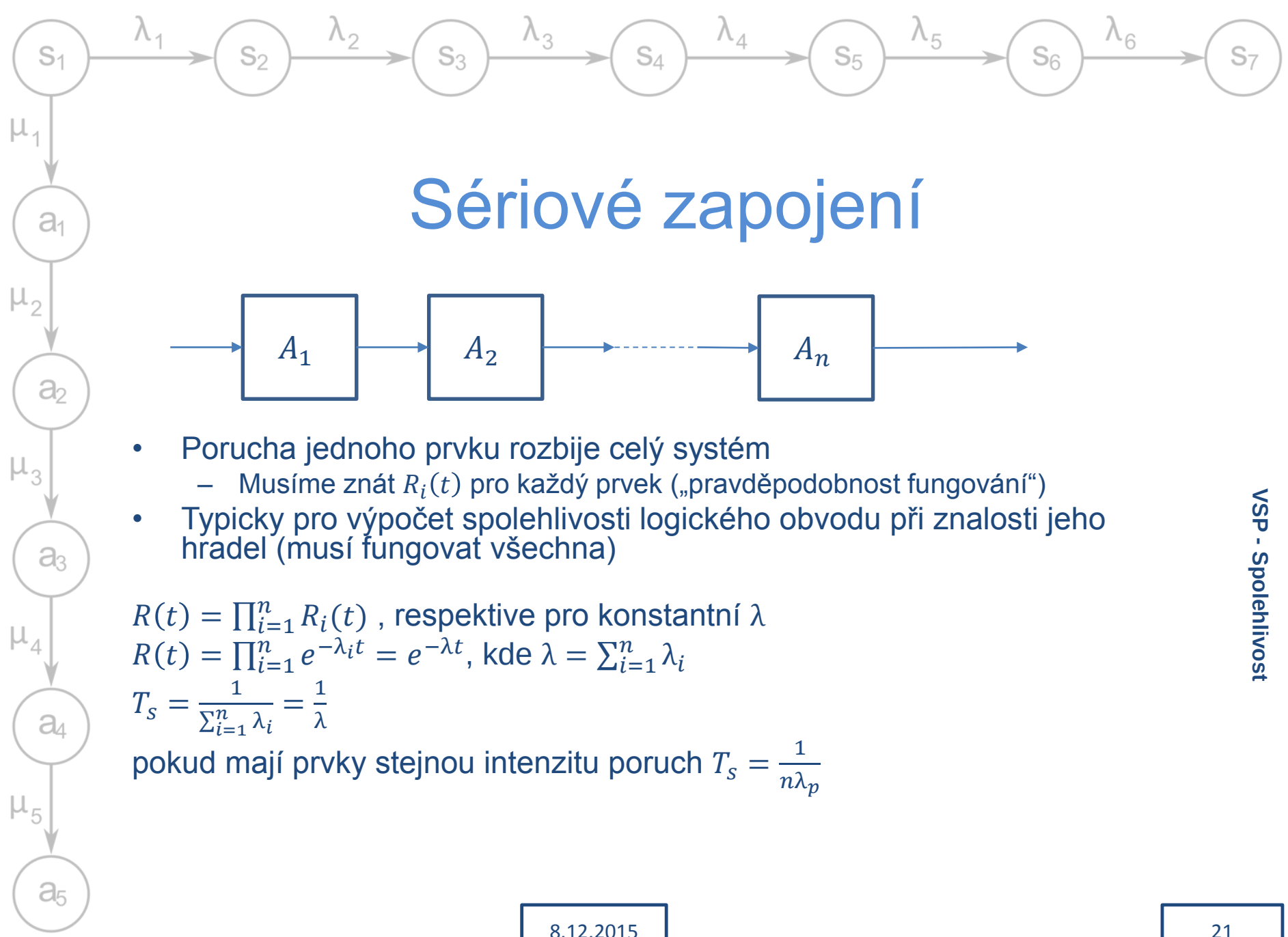


Systémy s nezávislými prvky

- Systém složený z více prvků se známými vlastnostmi (hradla, CPU a RAM, počítače, … - podle úrovně modelu)
- Poruchy jednotlivých prvků mohou být nezávislé
  - CPU se rozbije bez ohledu na to že je rozbitá paměť
  - Pozor, některé poruchy mohou podmínit vznik jiné poruchy, pak nelze mluvit o nezávislosti

  → **pak lze poruchy chápat jako nezávislé náhodné jevy**

- Typické pro neobnovované systémy
  - Oprava podmíněna předchozí poruchou
- Spojení prvků – logické, z hlediska spolehlivosti (nemusí být kopie reálného zapojení obvodu)



Sériové zapojení

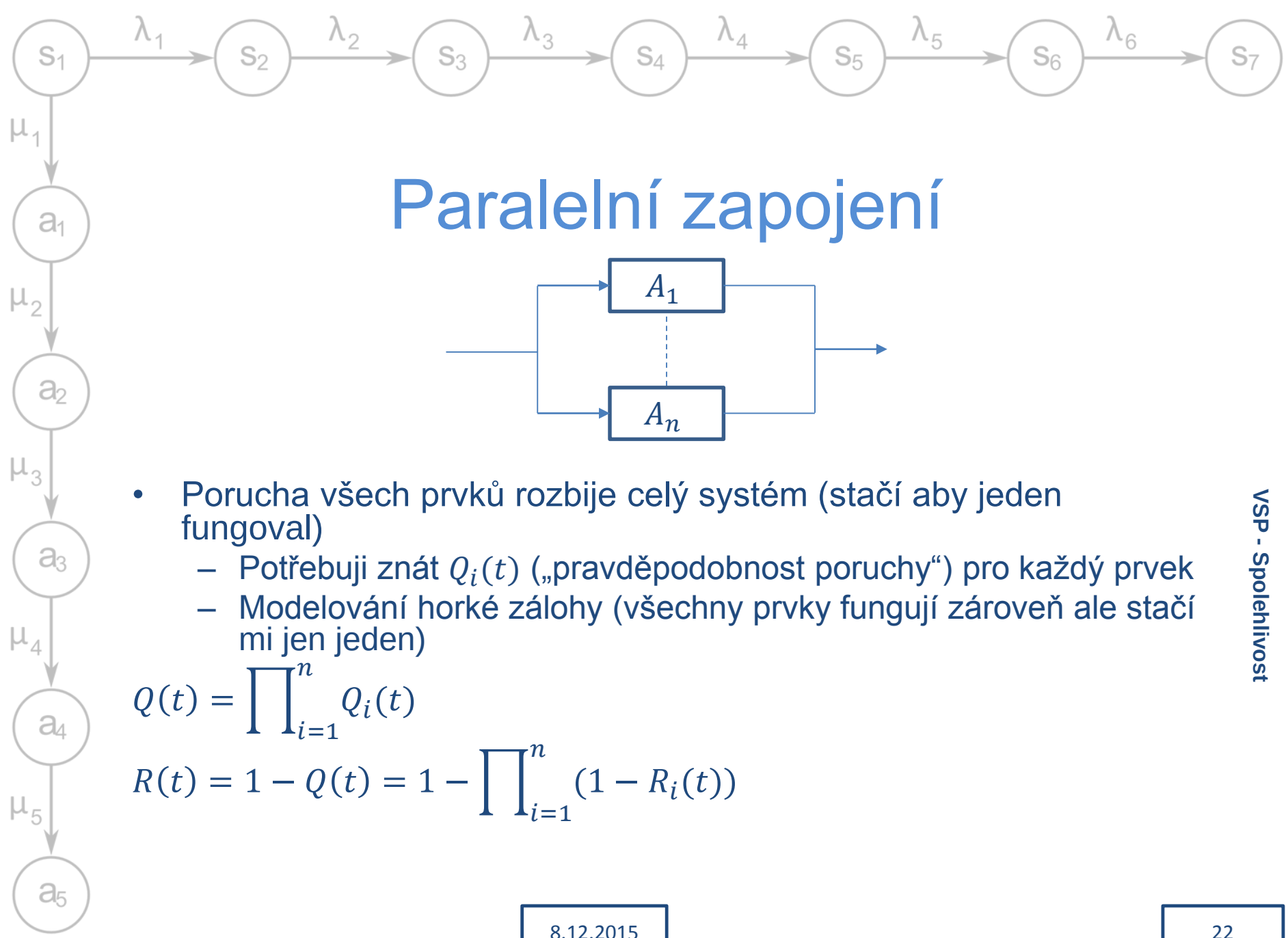
- Porucha jednoho prvku rozbije celý systém
  - Musíme znát $R_i(t)$ pro každý prvek („pravděpodobnost fungování“)
- Typicky pro výpočet spolehlivosti logického obvodu při znalosti jeho hradel (musí fungovat všechna)

$R(t) = \prod_{i=1}^{n} R_i(t)$, respektive pro konstantní λ

$R(t) = \prod_{i=1}^{n} e^{-\lambda_i t} = e^{-\lambda t}$, kde $\lambda = \sum_{i=1}^{n} \lambda_i$

$T_s = \frac{1}{\sum_{i=1}^{n} \lambda_i} = \frac{1}{\lambda}$

pokud mají prvky stejnou intenzitu poruch $T_s = \frac{1}{n\lambda_p}$

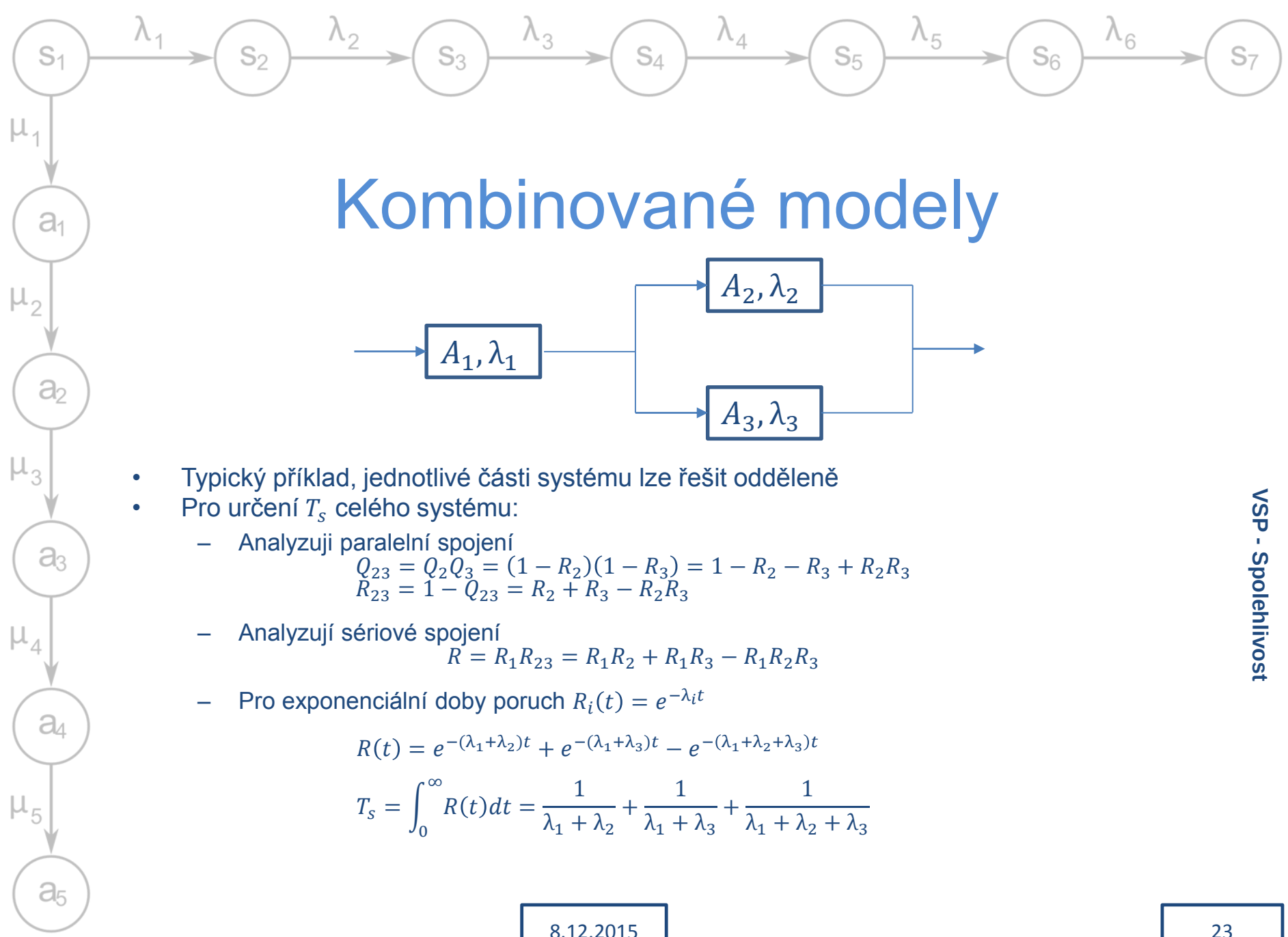


Paralelní zapojení

- Porucha všech prvků rozbije celý systém (stačí aby jeden fungoval)
  - Potřebuji znát $Q_i(t)$ („pravděpodobnost poruchy“) pro každý prvek
  - Modelování horké zálohy (všechny prvky fungují zároveň ale stačí mi jen jeden)

$$Q(t) = \prod_{i=1}^{n} Q_i(t)$$

$$R(t) = 1 - Q(t) = 1 - \prod_{i=1}^{n} (1 - R_i(t))$$



Kombinované modely

- Typický příklad, jednotlivé části systému lze řešit odděleně
- Pro určení T_s celého systému:
  - Analyzuji paralelní spojení

$$Q_{23} = Q_2 Q_3 = (1 - R_2)(1 - R_3) = 1 - R_2 - R_3 + R_2 R_3$$
$$R_{23} = 1 - Q_{23} = R_2 + R_3 - R_2 R_3$$

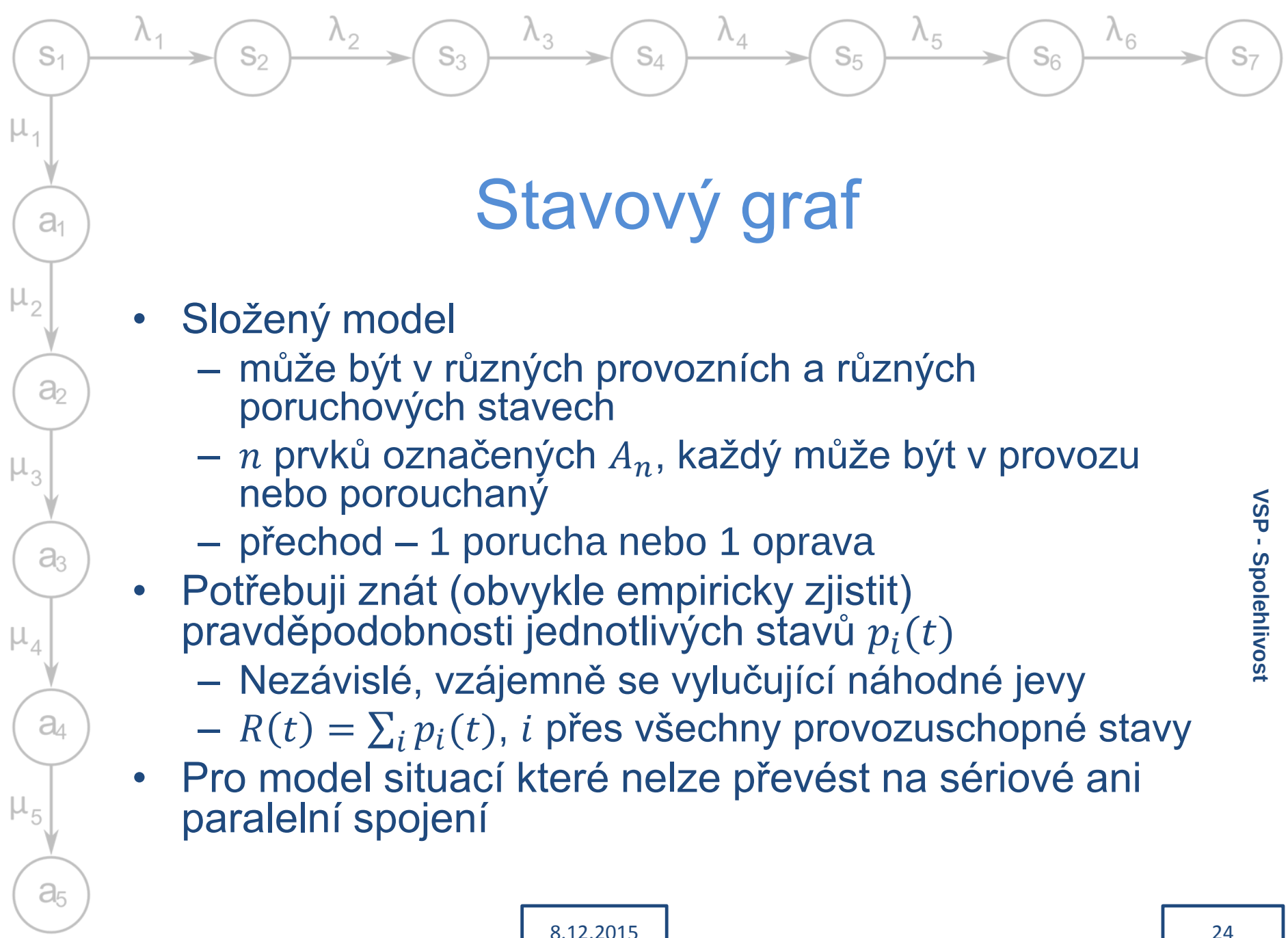
  - Analyzují sériové spojení

$$R = R_1 R_{23} = R_1 R_2 + R_1 R_3 - R_1 R_2 R_3$$

  - Pro exponenciální doby poruch $R_i(t) = e^{-\lambda_i t}$

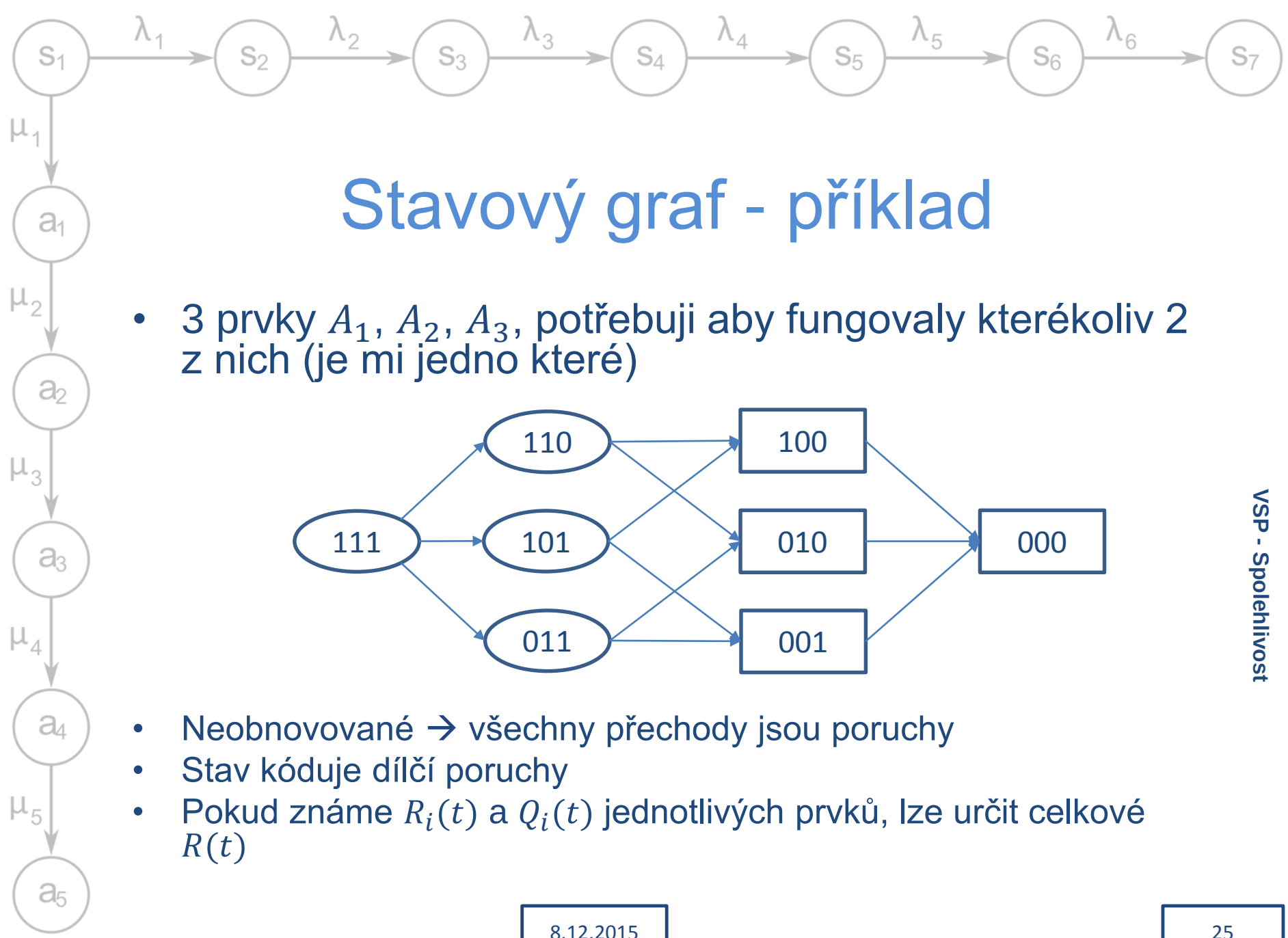
$$R(t) = e^{-(\lambda_1+\lambda_2)t} + e^{-(\lambda_1+\lambda_3)t} - e^{-(\lambda_1+\lambda_2+\lambda_3)t}$$

$$T_s = \int_0^\infty R(t)dt = \frac{1}{\lambda_1+\lambda_2} + \frac{1}{\lambda_1+\lambda_3} + \frac{1}{\lambda_1+\lambda_2+\lambda_3}$$



Stavový graf

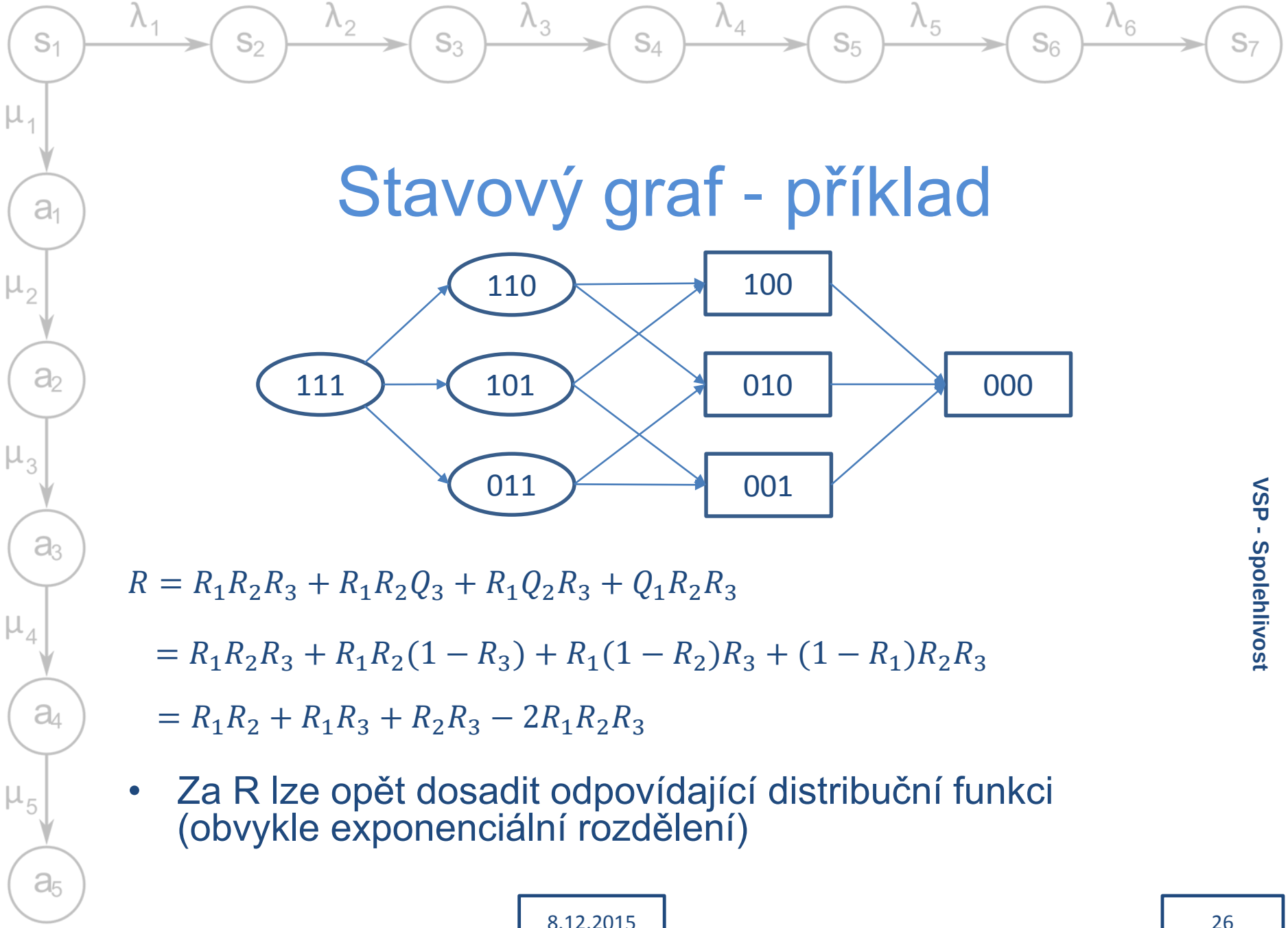
- Složený model
  - může být v různých provozních a různých poruchových stavech
  - n prvků označených A_n, každý může být v provozu nebo porouchaný
  - přechod – 1 porucha nebo 1 oprava
- Potřebuji znát (obvykle empiricky zjistit) pravděpodobnosti jednotlivých stavů $p_i(t)$
  - Nezávislé, vzájemně se vylučující náhodné jevy
  - $R(t) = \sum_i p_i(t)$, i přes všechny provozuschopné stavy
- Pro model situací které nelze převést na sériové ani paralelní spojení



Stavový graf - příklad

- 3 prvky A_1, A_2, A_3, potřebuji aby fungovaly kterékoliv 2 z nich (je mi jedno které)

- Neobnovované → všechny přechody jsou poruchy
- Stav kóduje dílčí poruchy
- Pokud známe $R_i(t)$ a $Q_i(t)$ jednotlivých prvků, lze určit celkové $R(t)$



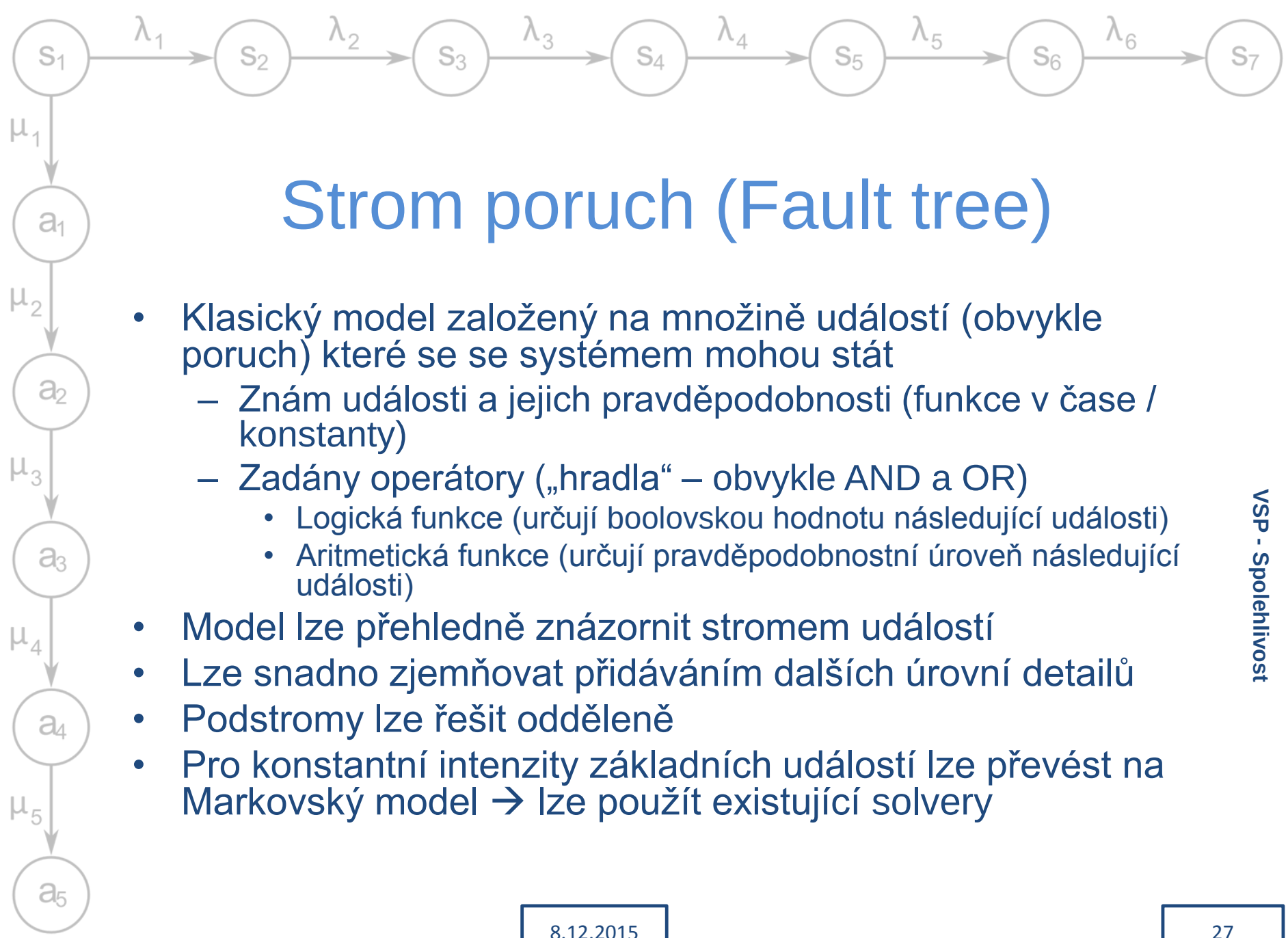
Stavový graf - příklad

$$R = R_1R_2R_3 + R_1R_2Q_3 + R_1Q_2R_3 + Q_1R_2R_3$$

$$= R_1R_2R_3 + R_1R_2(1 - R_3) + R_1(1 - R_2)R_3 + (1 - R_1)R_2R_3$$

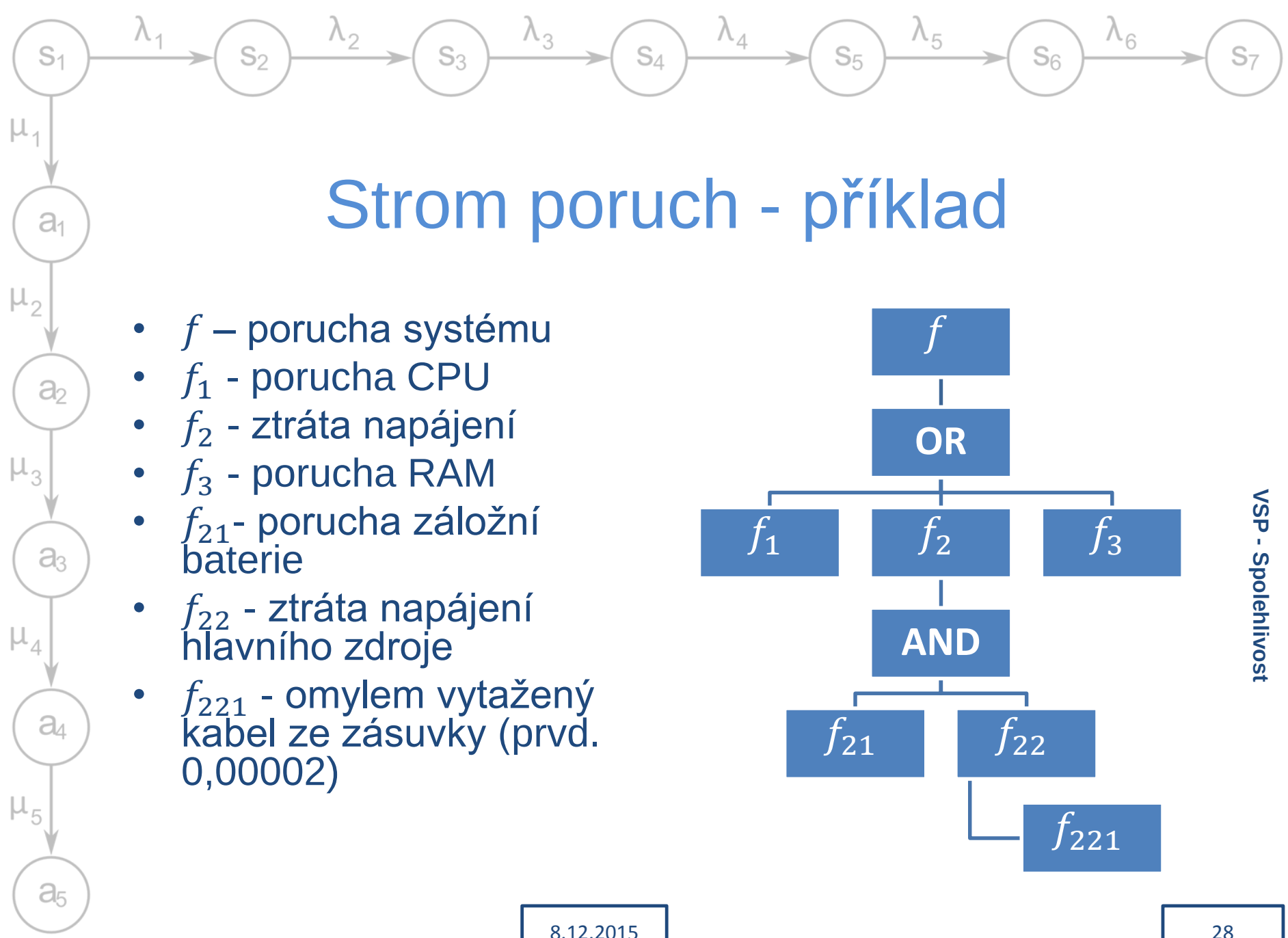
$$= R_1R_2 + R_1R_3 + R_2R_3 - 2R_1R_2R_3$$

- Za R lze opět dosadit odpovídající distribuční funkci (obvykle exponenciální rozdělení)



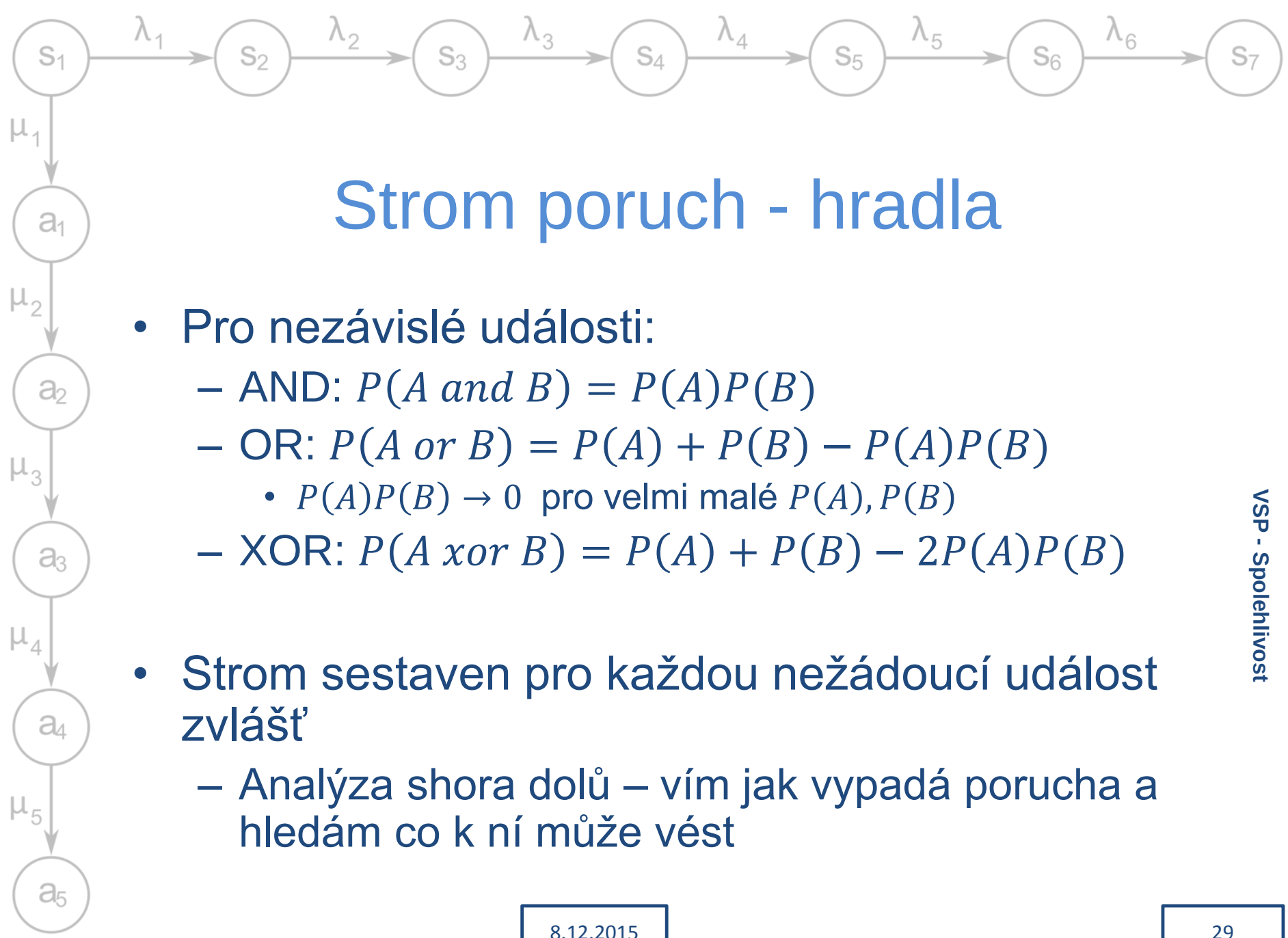
Strom poruch (Fault tree)

- Klasický model založený na množině událostí (obvykle poruch) které se se systémem mohou stát
  - Znám události a jejich pravděpodobnosti (funkce v čase / konstanty)
  - Zadány operátory („hradla“ – obvykle AND a OR)
    - Logická funkce (určují boolovskou hodnotu následující události)
    - Aritmetická funkce (určují pravděpodobnostní úroveň následující události)
- Model lze přehledně znázornit stromem událostí
- Lze snadno zjemňovat přidáváním dalších úrovní detailů
- Podstromy lze řešit odděleně
- Pro konstantní intenzity základních událostí lze převést na Markovský model → lze použít existující solvery



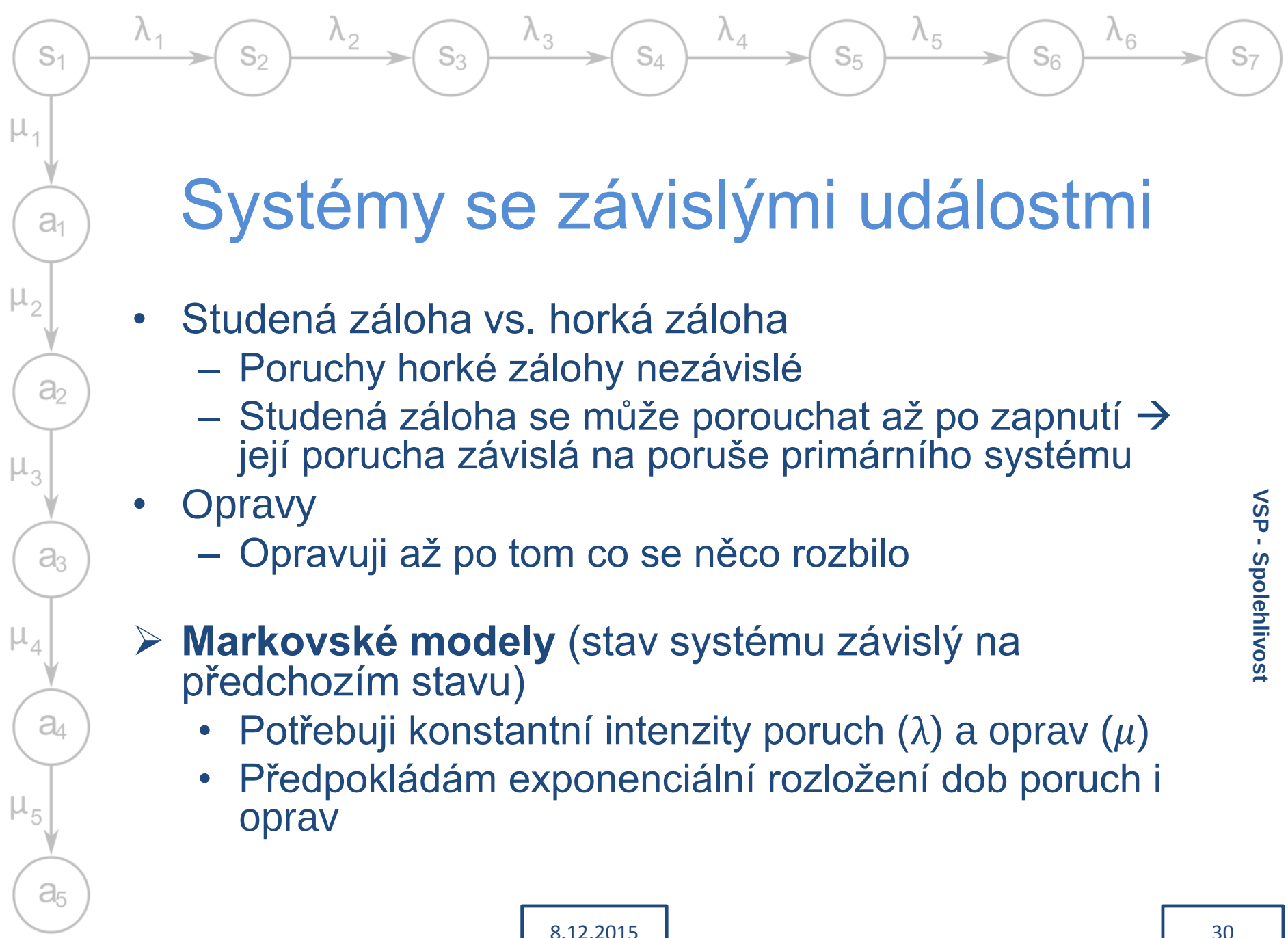
Strom poruch - příklad

- f – porucha systému
- f_1 - porucha CPU
- f_2 - ztráta napájení
- f_3 - porucha RAM
- f_{21}- porucha záložní baterie
- f_{22} - ztráta napájení hlavního zdroje
- f_{221} - omylem vytažený kabel ze zásuvky (prvd. 0,00002)



Strom poruch - hradla

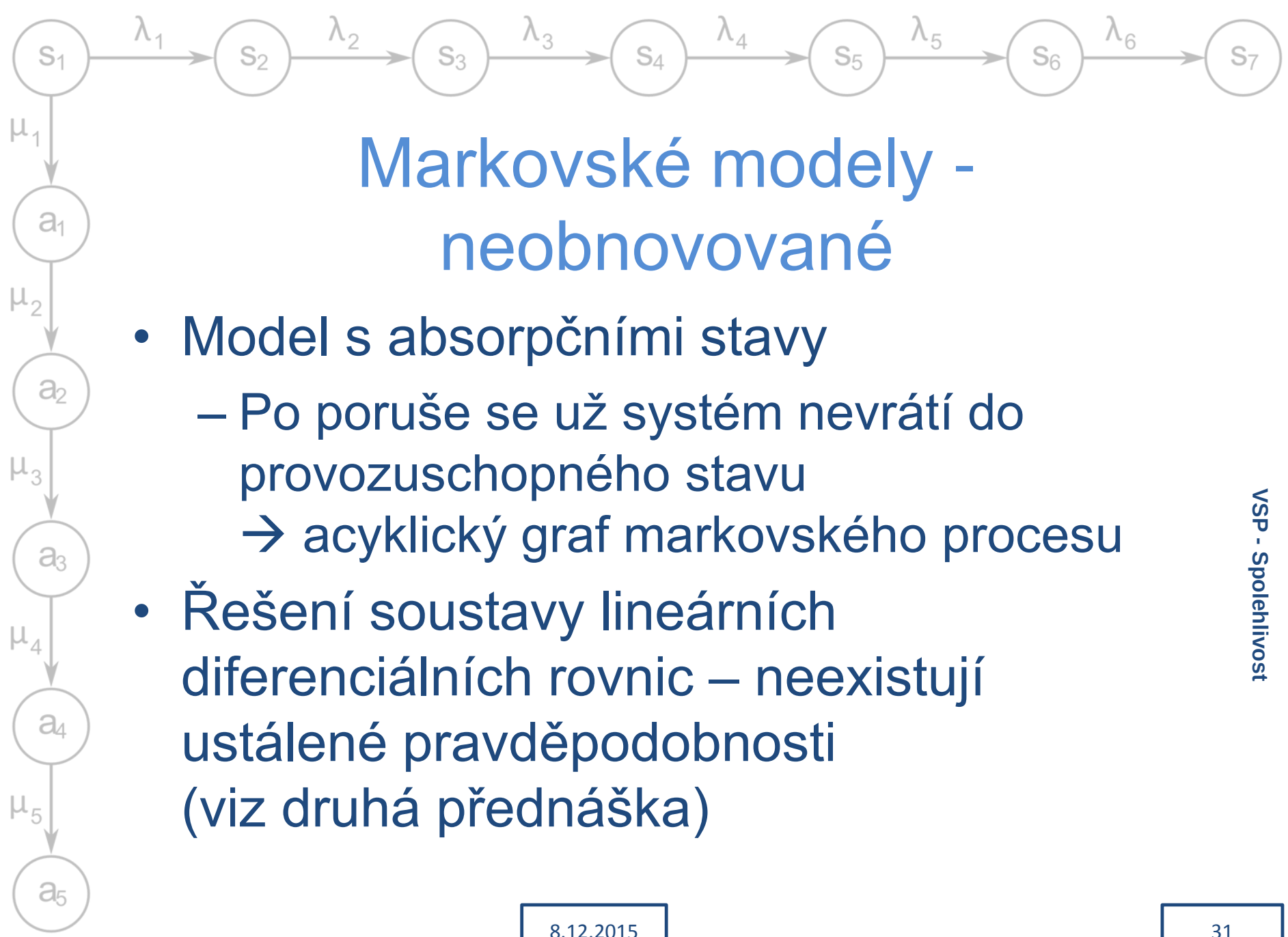
- Pro nezávislé události:
  - AND: $P(A\ and\ B) = P(A)P(B)$
  - OR: $P(A\ or\ B) = P(A) + P(B) - P(A)P(B)$
    - $P(A)P(B) \to 0$ pro velmi malé $P(A), P(B)$
  - XOR: $P(A\ xor\ B) = P(A) + P(B) - 2P(A)P(B)$
- Strom sestaven pro každou nežádoucí událost zvlášť
  - Analýza shora dolů – vím jak vypadá porucha a hledám co k ní může vést



Systémy se závislými událostmi

- Studená záloha vs. horká záloha
  - Poruchy horké zálohy nezávislé
  - Studená záloha se může porouchat až po zapnutí → její porucha závislá na poruše primárního systému
- Opravy
  - Opravuji až po tom co se něco rozbilo

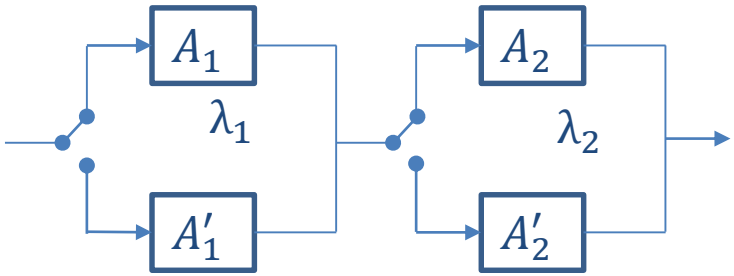
➢ **Markovské modely** (stav systému závislý na předchozím stavu)
- Potřebuji konstantní intenzity poruch (λ) a oprav (μ)
- Předpokládám exponenciální rozložení dob poruch i oprav

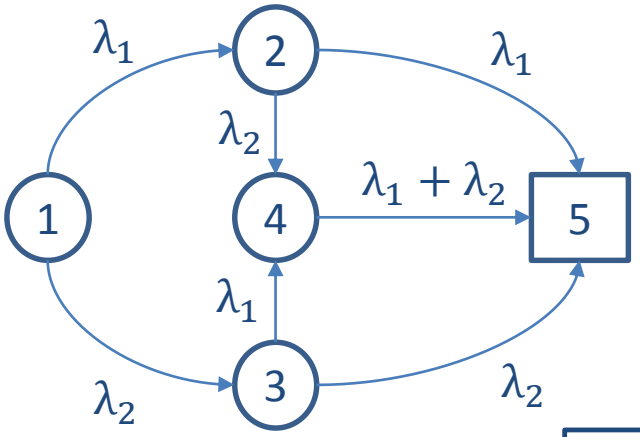


Markovské modely - neobnovované

- Model s absorpčními stavy
  - Po poruše se už systém nevrátí do provozuschopného stavu
    → acyklický graf markovského procesu
- Řešení soustavy lineárních diferenciálních rovnic – neexistují ustálené pravděpodobnosti (viz druhá přednáška)

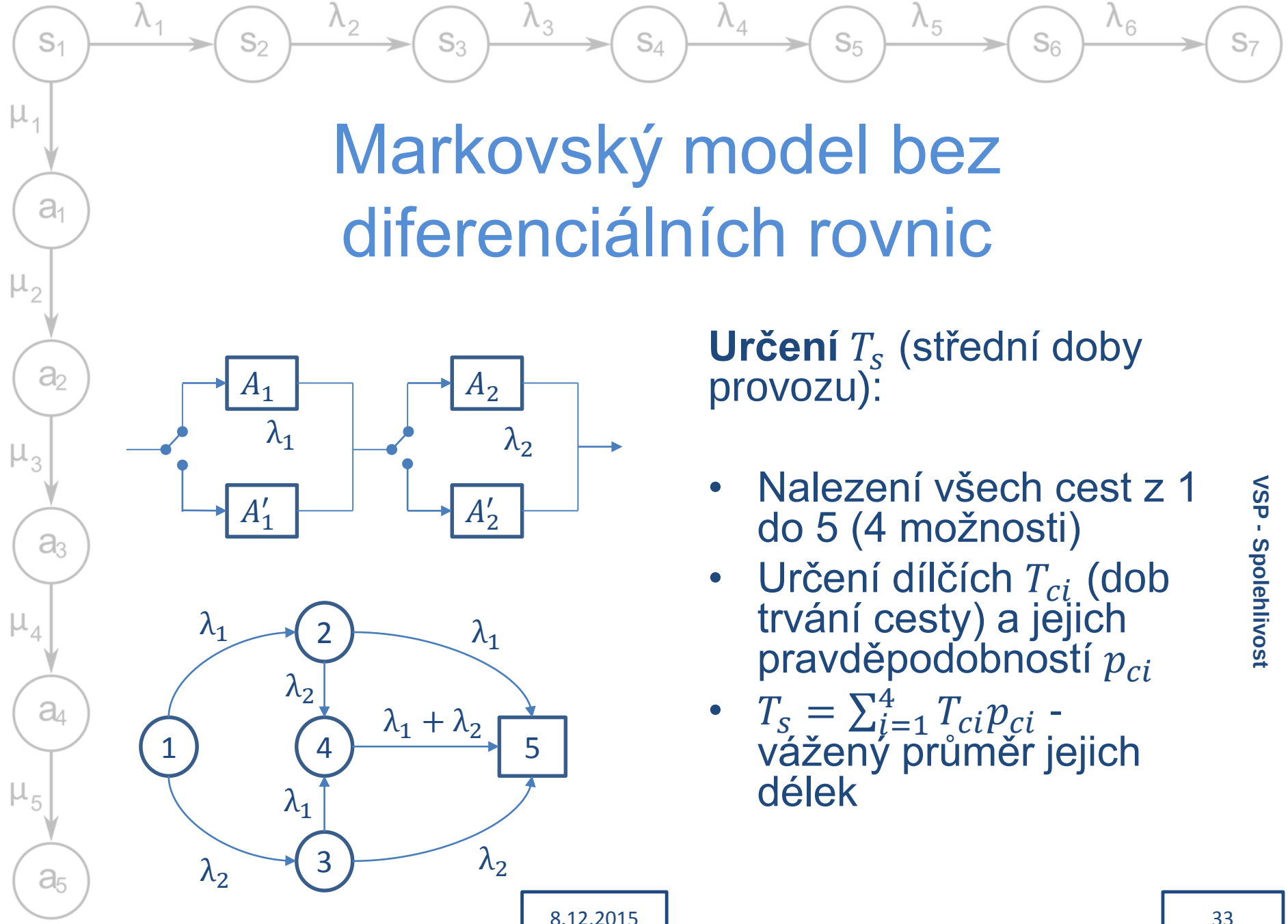
Markovský model bez diferenciálních rovnic





Stavy modelu:

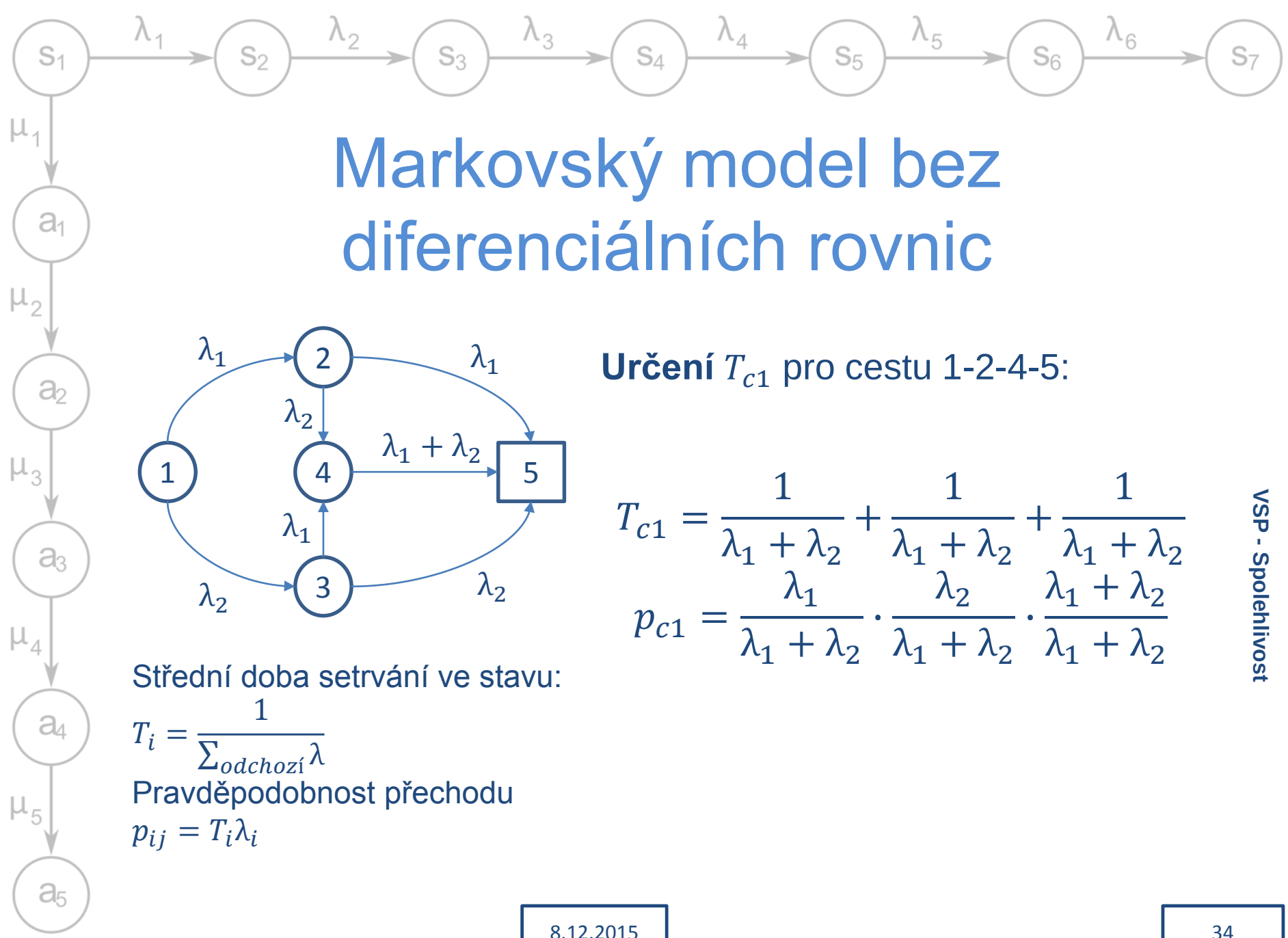
1. Vše v pořádku
2. Modul A_1 nahrazen zálohou
3. Modul A_2 nahrazen zálohou
4. Oba moduly nahrazeny zálohou
5. Systém je nefunkční



Markovský model bez diferenciálních rovnic

Určení T_s (střední doby provozu):

- Nalezení všech cest z 1 do 5 (4 možnosti)
- Určení dílčích T_{ci} (dob trvání cesty) a jejich pravděpodobností p_{ci}
- $T_s = \sum_{i=1}^{4} T_{ci} p_{ci}$ - vážený průměr jejich délek



Markovský model bez diferenciálních rovnic

Určení T_{c1} pro cestu 1-2-4-5:

$$T_{c1} = \frac{1}{\lambda_1 + \lambda_2} + \frac{1}{\lambda_1 + \lambda_2} + \frac{1}{\lambda_1 + \lambda_2}$$

$$p_{c1} = \frac{\lambda_1}{\lambda_1 + \lambda_2} \cdot \frac{\lambda_2}{\lambda_1 + \lambda_2} \cdot \frac{\lambda_1 + \lambda_2}{\lambda_1 + \lambda_2}$$

Střední doba setrvání ve stavu:

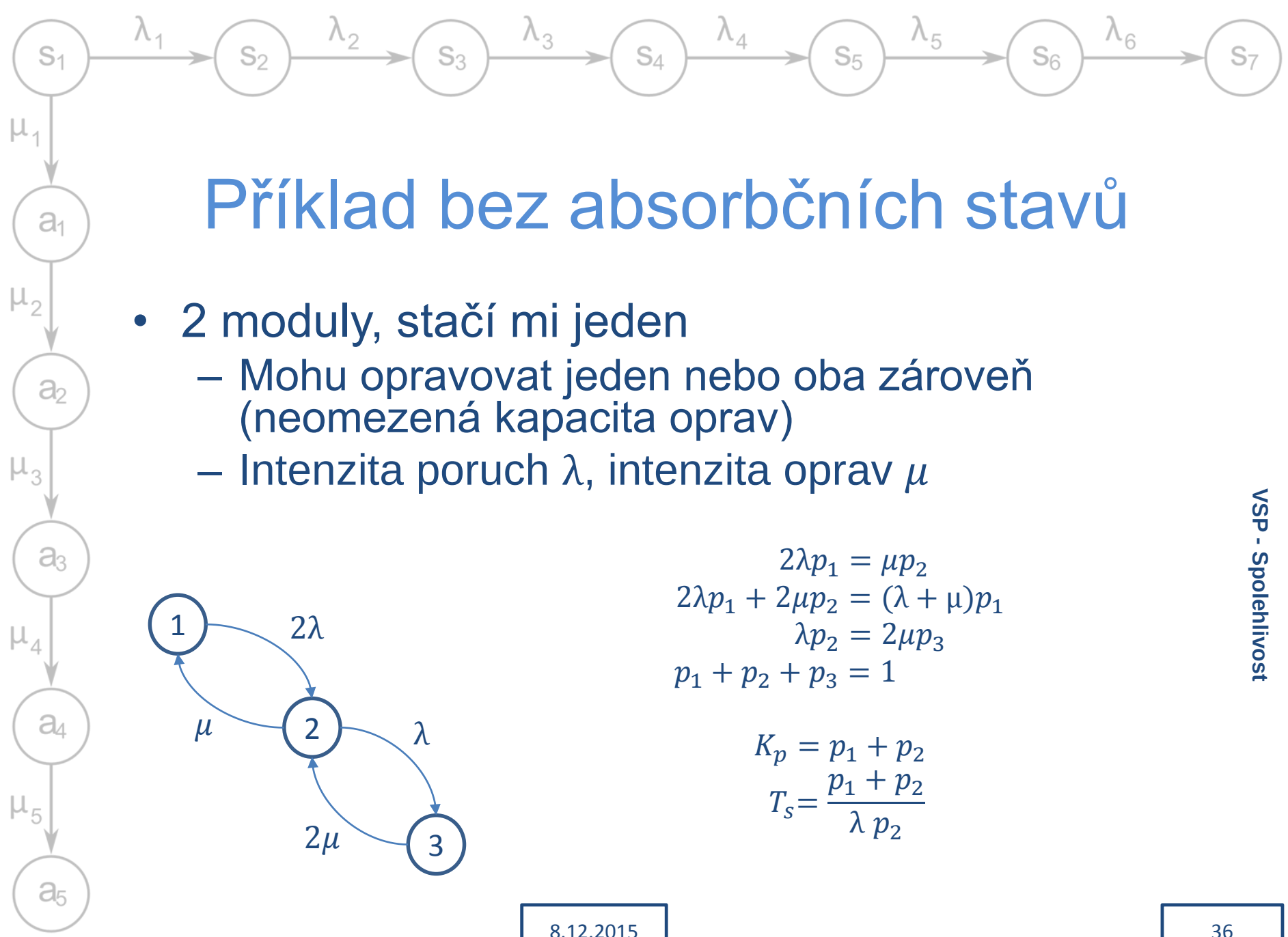
$$T_i = \frac{1}{\sum_{odchozí} \lambda}$$

Pravděpodobnost přechodu

$$p_{ij} = T_i \lambda_i$$

Markovské modely – obnovované

- Grafy obsahují cykly
  - Poruchy jedním směrem, opravy druhým
  - Stále mohou obsahovat absorbční stavy (něco opravit nejde)
  - Model pro nekonečně dlouhou dobu života → pokud jde vše opravit, lze pracovat s ustálenými pravděpodobnostmi
    - Místo diferenciálních rovnic dostanu lineární
- „*availability models*“ (místo „*reliability models*“)
- Koeficienty lze určit na základě analýzy pravděpodobnosti stavů modelu

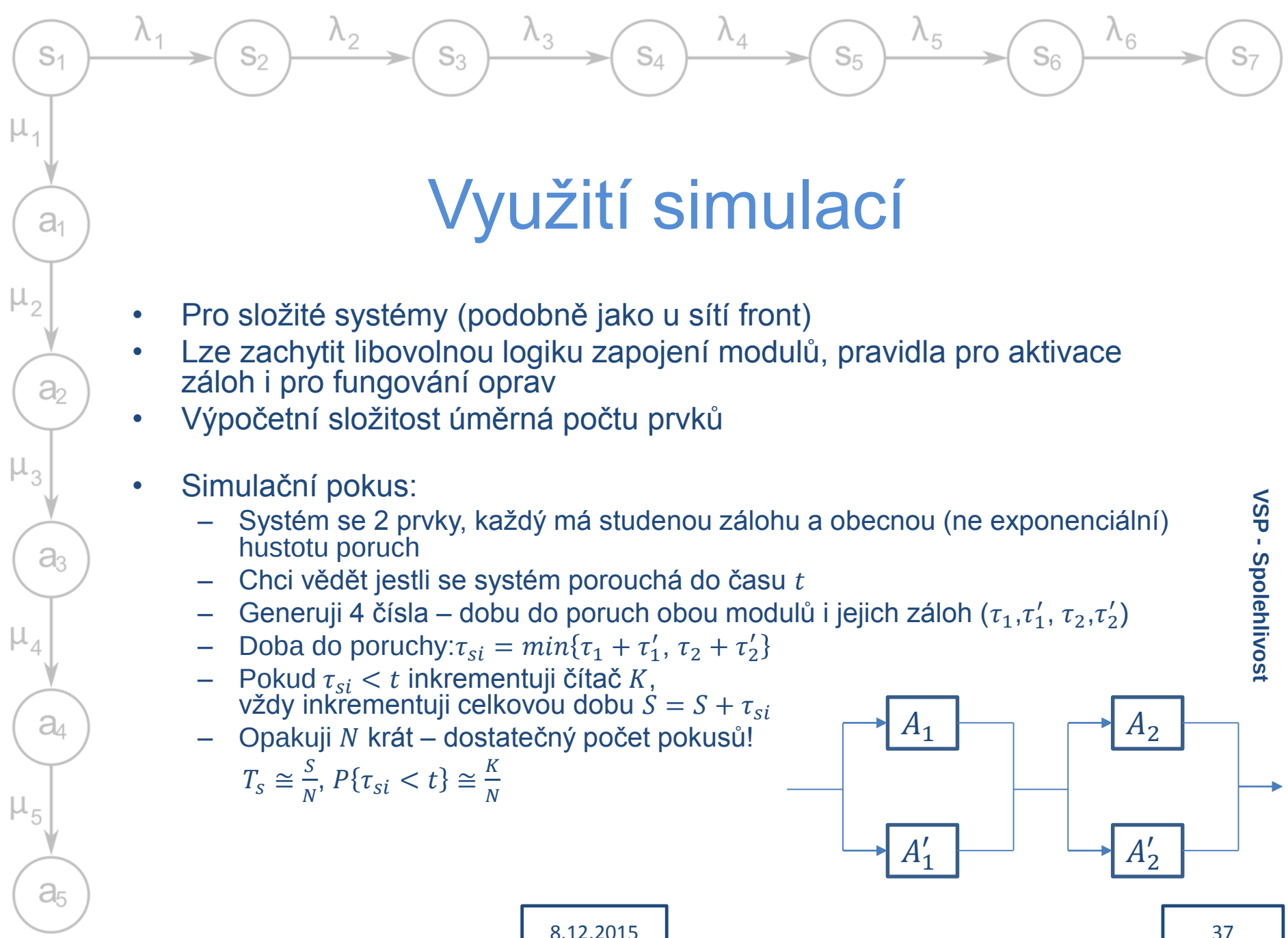


Příklad bez absorbčních stavů

- 2 moduly, stačí mi jeden
  - Mohu opravovat jeden nebo oba zároveň (neomezená kapacita oprav)
  - Intenzita poruch λ, intenzita oprav μ

$$2\lambda p_1 = \mu p_2$$
$$2\lambda p_1 + 2\mu p_2 = (\lambda + \mu) p_1$$
$$\lambda p_2 = 2\mu p_3$$
$$p_1 + p_2 + p_3 = 1$$

$$K_p = p_1 + p_2$$
$$T_s = \frac{p_1 + p_2}{\lambda\, p_2}$$

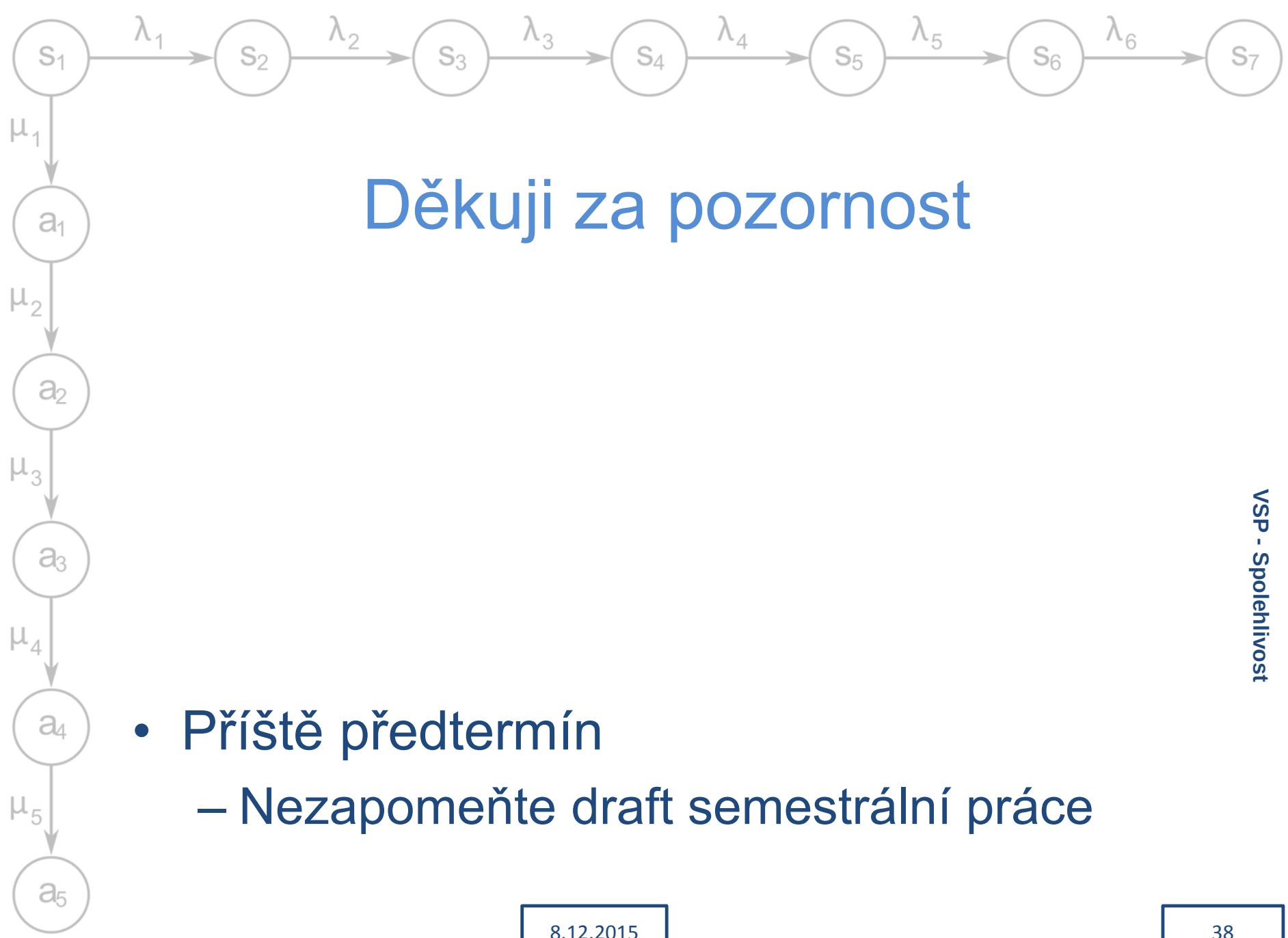


Využití simulací

- Pro složité systémy (podobně jako u sítí front)
- Lze zachytit libovolnou logiku zapojení modulů, pravidla pro aktivace záloh i pro fungování oprav
- Výpočetní složitost úměrná počtu prvků

- Simulační pokus:
  - Systém se 2 prvky, každý má studenou zálohu a obecnou (ne exponenciální) hustotu poruch
  - Chci vědět jestli se systém porouchá do času t
  - Generuji 4 čísla – dobu do poruch obou modulů i jejich záloh (τ_1,τ_1', τ_2,τ_2')
  - Doba do poruchy:$\tau_{si} = min\{\tau_1 + \tau_1', \tau_2 + \tau_2'\}$
  - Pokud $\tau_{si} < t$ inkrementuji čítač K,
    vždy inkrementuji celkovou dobu $S = S + \tau_{si}$
  - Opakuji N krát – dostatečný počet pokusů!

    $T_s \cong \frac{S}{N}$, $P\{\tau_{si} < t\} \cong \frac{K}{N}$



Děkuji za pozornost

- Příště předtermín
  - Nezapomeňte draft semestrální práce