

5

Estratégias de Detecção Sensíveis à História

O objetivo dessas estratégias é realizar a detecção de elementos do código considerando algum comportamento evolutivo do mesmo. Nesse sentido, estratégias de detecção sensíveis à história devem conter obrigatoriamente pelo menos uma métrica sensível à história. Métricas introduzem informações quantitativas enquanto estratégias de detecção complementam com algum raciocínio qualitativo. Figueiredo (2006), ao propor estratégias para avaliar o espalhamento de interesses no código, apresenta inicialmente um conjunto de regras (ou estratégias) formadas por apenas uma métrica e apresenta em seguida as regras mais complexas. A abordagem utilizada por esse autor nos inspira na apresentação de nossas estratégias.

A utilização das métricas sensíveis à história (Seção 4) quando disassociadas de estratégias de detecção também geram uma grande quantidade de resultados possivelmente relativos a um grande número de componentes do sistema. Assim como ocorre no caso de métricas convencionais, isso dificulta a identificação de todos os componentes que apresentam valores anormais de métricas. Estratégias simples de detecção sensíveis à história podem ser utilizadas para recuperar uma família de componentes caracterizados pelo mesmo comportamento evolutivo. Por exemplo, coesão decrescente ao longo das versões, acoplamento crescente, dentre outros. Para casos como esses, estratégias podem ser formuladas através de uma única métrica SH. Como outra possibilidade, estratégias de detecção SHs mais complexas podem ser utilizadas na investigação de anomalias de modularidade clássicas como as apresentadas em (Fowler et al. 1999).

Neste capítulo apresentamos alguns exemplos de detecções simples que objetivam a recuperação de componentes cuja propriedades de código apresentam o mesmo comportamento evolutivo (Seção 5.2). Em seguida apresentamos estratégias mais complexas a serem utilizadas em detecções de anomalias de modularidade (Fowler et al. 1999) (Seção 5.3). Enquanto as primeiras ajudam na interpretação dos resultado de uma única métrica, as últimas ajudam na interpretação de uma combinação de métricas que visam capturar sintomas de anomalias clássicas. Antes da apresentação dessas estratégias realizamos uma

breve explicação sobre a utilização de valores limites, uma questão ainda bastante discutida quando se trabalha com recursos baseados em medição (Seção 5.1).

5.1

A Questão dos Valores Limites

Antes de começar a apresentação das estratégias de detecção é importante destacar um aspecto que tem influência decisiva na classificação dos problemas detectados por estratégias de detecção: os valores limites. Essa questão, longe de ser nova, caracteriza intrinsecamente qualquer abordagem baseada em métricas. Duas estratégias podem ser usadas na definição de valores limites: (1) recomendações previamente documentadas e (2) configuração empírica. No primeiro caso, a escolha do valor limite é guiada por experiências semelhantes do passado e recomendações de autores de trabalhos relacionados a medições (Lorenz e Kidd 1994).

Entretanto, quando essas recomendações não estão disponíveis, então os valores podem ser configurados de acordo com características do sistema (por exemplo, tamanho) e avaliados em estudos empíricos. Por exemplo, se um determinado valor gerar mais falsos positivos ou falsos negativos do que resultados corretos, este valor pode ser alterado e reavaliado. Na verdade, é por isso que, quando uma ferramenta de suporte a estratégias se encontra disponível, é importante que ajustes de valores limites sejam facilitados. O desenvolvedor pode definir um valor e verificar imediatamente os efeitos desta ação (alertas excessivos ou escassos); variando continuamente este valor até restringir os alertas a um número que pode ser verificado em tempo hábil e verificar se os alertas gerados estão sendo efetivos (Figueiredo 2006).

Os valores limites da maioria das estratégias sensíveis à história propostas serão representados por letras gregas. Isso porque um considerável número de estudos empíricos seria necessário para que pudéssemos sugerir tais limites. Nesse momento, extrapola os objetivos desta dissertação realizar estudos empíricos para a sugestão de valores limite. À princípio, como usualmente ocorre, caberá ao desenvolvedor especificar quais valores serão utilizados para parametrizar tais estratégias. Para isso, a ferramenta de medição e avaliação proposta nesse trabalho (Capítulo 6) fornece a flexibilidade necessária para que ajustes sejam feitos. No processo de avaliação das estratégias SHs foram utilizadas ambas as formas para definição de valores limites, (1) sugestões de literatura relacionada e (2) configuração empírica. Já no caso das estratégias convencionais apenas a forma (1) foi utilizada, seguindo os valores sugeridos pelos autores das estratégias.

5.2 Detecções Simples

Através da utilização direta das métricas propostas no Capítulo 4, algumas estratégias bastante simples podem ser elaboradas. Através dessas estratégias, filtros podem ser especificados possibilitando ao desenvolvedor a recuperação de um grupo de componentes com o mesmo comportamento evolutivo. Estratégias desse tipo podem alertar o desenvolvedor, por exemplo, sobre decréscimo ou crescimento de propriedades importantes em desenvolvimento de sistemas, como coesão, complexidade, tamanho, ou outras. A seguir, apresentamos algumas sugestões de estratégias simples que podem ser realizadas pela aplicação direta de uma única métrica sensível à história.

1. Acoplamento Altamente Crescente: componentes em que o acoplamento teve crescimento muito elevado em relação ao projeto inicial do sistema. A degeneração da propriedade é avaliada apenas em relação ao acoplamento na primeira versão do componente.
2. Acoplamento Estritamente Crescente: componentes em que o acoplamento se manteve crescente em todas as versões pertencentes ao histórico do sistema.
3. Coesão Altamente Decrescente: componentes em que o decréscimo da coesão apresenta-se muito elevado em relação ao projeto inicial do sistema. A degeneração da propriedade é avaliada apenas em relação à coesão na primeira versão do componente.
4. Coesão Estritamente Decrescente: componentes em que a coesão se manteve decrescente em todas as versões pertencentes ao histórico do sistema.
5. Tamanho Altamente Decrescente: componentes em que o tamanho teve crescimento muito elevado em relação ao projeto inicial do sistema. A degeneração da propriedade é avaliada apenas em relação ao tamanho medido na primeira versão do componente.
6. Tamanho Estritamente Crescente: componentes em que o tamanho se manteve crescente em todas as versões pertencentes ao histórico do sistema.
7. Tamanho de Elevada Instabilidade: componentes em que o tamanho foi alterado em todas as versões pertencentes ao histórico do sistema.

8. Tamanho Constante: componentes em que o tamanho não sofreu alterações em nenhuma das versões pertencentes ao histórico do sistema.

As estratégias de número 1 e 5 avaliam o crescimento de acoplamento e tamanho em relação ao projeto inicial dos componentes. Tal avaliação pode ser suportada por métricas SH do tipo *gpi*. Na estratégia nº1, pode ser usada a *gpiCBO* (Aumento Percentual Geral de CBO) e na nº5, a *gpiLOC* (Aumento Percentual Geral de LOC). Na listagem também são sugeridas duas estratégias para recuperar componentes com propriedades estritamente crescentes ao longo das versões do sistema (nº2 e nº6). Nesse caso, podem ser utilizadas métricas SH do tipo *rni*, *rniCBO* (Número Relativo de Crescimento do CBO) na nº2, e *rniLOC* (Número Relativo de Crescimento do LOC) na nº6.

Nas estratégias de nº3 e 4, que consideram a propriedade coesão, podem ser utilizadas as métricas *gpdTCC* (Decrescimento Percentual Geral de TCC) e *rndTCC* (Número Relativo de Decrescimento de TCC), respectivamente. A estratégia nº7, que considera a elevada instabilidade relativa ao tamanho de um componente, pode ser suportada pela métrica *rncLOC* (Número Relativo de Alterações em LOC). Já na estratégia nº8, que se refere à constância do tamanho em todo o histórico do componente, pode ser utilizada a *ancLOC* (Número Absoluto de Alterações em LOC). Os tipos de detecções simples sugeridas bem como as métricas de suporte mencionadas são apresentadas na Tabela 5.1.

Detecções Relativas à Evolução de Acoplamento	
1. Acoplamento Altamente Crescente	$gpiCBO > \alpha$
2. Acoplamento Estritamente Crescente	$rniCBO = 1$
Detecções Relativas à Evolução da Coesão	
3. Coesão Altamente Decrescente	$gpdTCC > \alpha$
4. Coesão Estritamente Decrescente	$rndTCC = 1$
Detecções Relativas à Evolução do Tamanho	
5. Tamanho Altamente Crescente	$gpiLOC > \alpha$
6. Tamanho Estritamente Crescente	$rniLOC = 1$
7. Tamanho de Elevada Instabilidade	$rncLOC = 1$
8. Tamanho Constante	$ancLOC = 0$

Tabela 5.1: Exemplos de estratégias sensíveis à história com uma única métrica: estratégias simples

A utilização dos limites de valor 1 ou 0 (estratégias de nº2, 4, 6, 7 e 8) pode ser compreendida através das explicações das métricas de tipo *rni*, *rnd*, *rnc* e *anc* (Seção 4). Já os limites representados por letras gregas significa que esses limites devem ser configurados pelo usuário. Esses foram utilizados nas

estratégias nº1, 3 e 5 que avaliam a degeneração das propriedades acoplamento e tamanho em relação ao projeto inicial do componente. Daí a necessidade do usuário configurar o que ele considera como elevado valor de degeneração, em cada caso particular.

5.3 Detecções de Anomalias de Modularidade

Nesta seção, são consideradas estratégias mais complexas utilizadas para a identificação de anomalias de modularidade. Como mencionado na Seção 3.2, estratégias de detecção convencionais não consideram a evolução das propriedades do código. Além disso, não é difícil encontrar trabalhos que destaquem a baixa eficácia comumente obtida por essas estratégias. Com o objetivo de contribuir com detecções eficazes de anomalias de modularidade, iniciamos um estudo relacionado à proposição e avaliação de estratégias de detecção sensíveis à história para tal finalidade. Os problemas a serem detectados pelas estratégias são: *God Class* (GC), *Shotgun Surgery* (SS) e *Divergent Change* (DC) (Fowler et al. 1999, Riel 1996).

Para apresentação dessa seção seguiremos a seguinte metodologia. Para cada anomalia apresentaremos: (i) uma breve definição da anomalia, (ii) a estratégia convencional normalmente utilizada para realizar tal detecção e (iii) uma proposta de estratégia sensível a história. A apresentação da estratégia convencional visa facilitar possíveis comparações. Todas as estratégias SHs foram resultantes de análises empíricas dessas anomalias e suposições as quais tínhamos interesse de testar. Essas estratégias nos serviram como objetos de avaliação sobre as possíveis contribuições de estratégias sensíveis à história na detecção de anomalias de código.

Diversas configurações de estratégias sensíveis à história poderiam ser propostas e testadas neste trabalho, sejam elas puras ou híbridas. Por estratégias híbridas consideramos aquelas que são formadas tanto por métricas convencionais quanto por métricas sensíveis à história. Por estratégias sensíveis à história puras consideramos aquelas formadas apenas por métricas sensíveis à história. Essas diferentes configurações de estratégias são disponibilizadas na ferramenta proposta (Capítulo 6). Todas as estratégias apresentadas a seguir foram avaliadas e os resultados dessa avaliação são apresentados no Capítulo 7 deste trabalho.

5.3.1

God Class (GC)

Em um bom projeto OO, o trabalho do sistema deve tender a ser uniformemente distribuído entre as classes que o compõe. O problema de modularidade *God Class* se refere a classes que tendem a centralizar a maioria do trabalho de um sistema (Riel 1996). Ao fazer isso, uma instância de uma *God Class* normalmente apresenta-se grande e complexa, pois possui muitas responsabilidades ao mesmo tempo.

5.3.2 Estratégia de Detecção Convencional

A estratégia convencional de GCs sugerida por (Lanza e Marinescu 2006) é baseada em três características prováveis em classes desse tipo. Para detectar tal anomalia, esses autores sugerem que, primeiramente, sejam detectadas as classes que dependem fortemente dos dados provenientes de outras classes. Depois, o conjunto de classes detectadas é reduzido, eliminando as classes de baixa complexidade e com alta coesão. A partir desses passos, foi proposta a estratégia que é apresentada pela Figura 5.1.

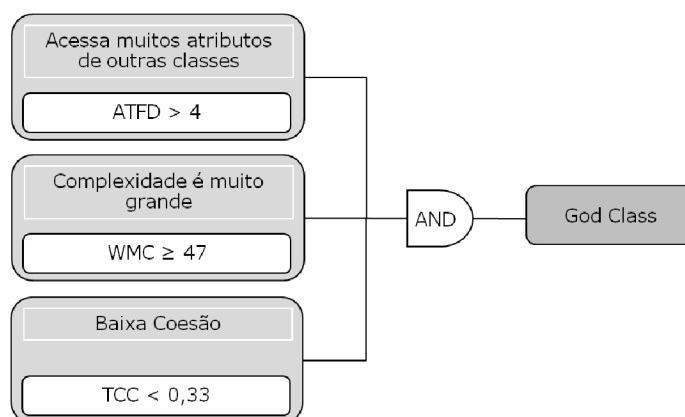


Figura 5.1: Estratégia de detecção convencional para *God Class*

As métricas e valores limites apresentados são sugeridos pelos próprios autores que se basearam em estudos empíricos para realizar tais sugestões. As métricas convencionais ATFD, WMC e TCC avaliam as propriedades de acoplamento, complexidade e coesão, respectivamente. Cada uma das métricas convencionais citadas nessa estratégia estão definidas no Apêndice A desta dissertação.

5.3.3 Estratégia de Detecção Sensível à História

A partir das informações normalmente apresentadas sobre esse problema e da observação de instâncias dessa anomalia, propusemos uma estratégia

sensível à história para avaliar as chances de detecção de GCs. Tal estratégia foi proposta pensando em testar se a captura de classes com complexidade e tamanho crescentes não seria capaz de detectar classes com tal anomalia. O raciocínio é baseado no fato de que se classes com tal anomalia tendem a acumular muitas responsabilidades ao mesmo tempo, é bastante provável que elas venham a acumular mais e mais responsabilidades ao longo das versões. Ou seja é possível que tais classes, também referenciadas como classes Bolha, tenham a tendência de “inchar” ao longo das versões, apresentando tamanho e complexidade crescentes. É o que investigaremos através da estratégia apresentada pela Figura 5.2 a seguir.

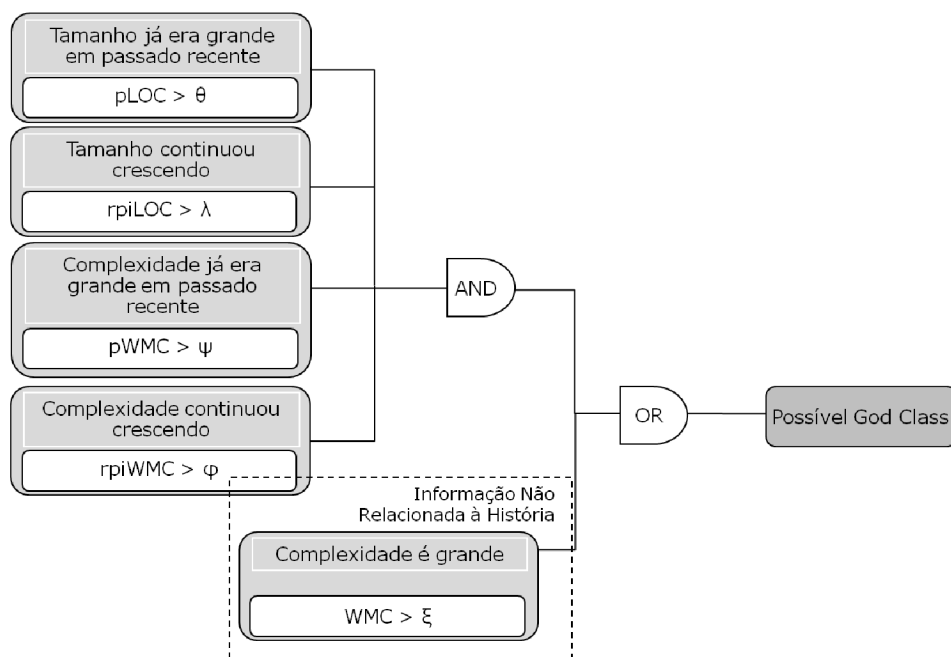


Figura 5.2: Estratégia sensível à história para *God Class*

Uma alternativa à utilização de LOC na avaliação do tamanho seria a utilização da métrica NOS que sofre menos influência de padrões de formatação no código. Entretanto, essa métrica é pouco suportada pelas ferramentas de medição, como vimos na Seção 3.1.6. A última cláusula que integra a estratégia, a destaca como *informação não sensível à história*, foi utilizada para possibilitar a realização e detecções mesmo em sistemas de uma única versão. Se a estratégia fosse composta apenas pelos quatro filtros primeiramente apresentados, como todos capturam apenas informações sensíveis à história, tal estratégia não possibilitaria detecções na versão inicial no sistema, uma vez que a primeira versão estaria desprovida de histórico.

De forma sintetizada, a estratégia SH para GC apresenta o seguinte raciocínio:

Estratégia SH Híbrida: **SE** tamanho e complexidade já eram grandes em passado recente e tais propriedades continuam crescendo na versão atual
ENTÃO POSSÍVEL *God Class*

5.3.4

Shotgun Surgery

O problema de modularidade *Shotgun Surgery* corresponde a classes cuja modificação implica muitas pequenas modificações em muitas outras classes (Fowler et al. 1999). Quando as modificações estão espalhadas, elas são difíceis de encontrar e, por isso, é muito provável esquecer uma modificação importante. Isso acarreta, portanto, um impacto negativo na manutenibilidade dos sistemas já que as mudanças não estão agrupadas.

5.3.5

Estratégia de Detecção Convencional

Lanza e Marinescu (2006) na elaboração de uma estratégia de convencional para detectar a anomalia *Shotgun Surgery*, consideram que tais classes normalmente apresentam métodos que são muito utilizados por outras classes e que, além disso, elas possuem muitas outras classes como clientes. É o que é expresso pela estratégia representada pela Figura 5.3.

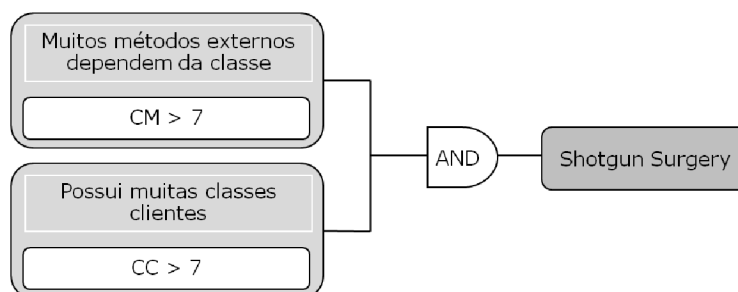


Figura 5.3: Estratégia de detecção convencional para *Shotgun Surgery*

As métricas CM e CC avaliam respectivamente a relação de acesso de outras classes aos métodos da classe avaliada e a quantidade de classes que potencialmente são afetadas mediante sua alteração. Cada uma das métricas convencionais citadas nessa estratégia estão apresentadas no Apêndice A desta dissertação.

5.3.6

Estratégia de Detecção Sensível à História

Para a detecção de *Shotgun Surgery* vamos tentar analisar a contribuição de informação sensível à história imbutida exatamente na mesma estratégia

convencional apresentada anteriormente para tal anomalia. Nesse caso, realizamos uma extensão da estratégia convencional, reaproveitando as mesmas métricas e valores limites e adicionamos uma única métrica sensível à história. Através dessa abordagem, pretendemos verificar exatamente que tipos de influência essa métrica sensível à história terá sobre os resultados da estratégia convencional proposta pela literatura. A utilização dessa única informação sensível à história é destacada na Figura 5.4 que representa a estratégia.

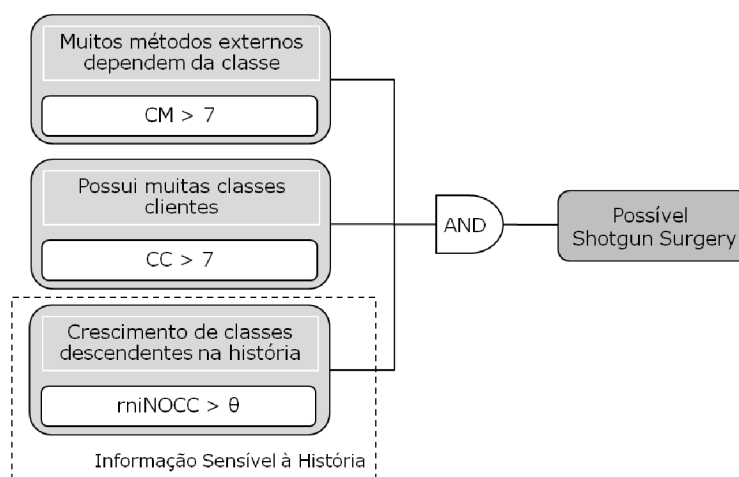


Figura 5.4: Estratégia sensível à história para *Shotgun Surgery*

A métrica sensível à história utilizada para estender a estratégia convencional foi a rniNOCC relacionada ao aumento de classes descendentes da classe avaliada. A utilização de uma métrica relacionada a hierarquia de classes, é justificada pelo seguinte raciocínio: se cresce o número de descendentes, também cresce a influência da classe avaliada sobre outras classes. Ou seja, é bastante provável que alterações em uma classe herdada por muitas outras possam resultar em alterações em cascata em seus descendentes (Chidamber e Kemerer 1994).

De forma sintetizada, a estratégia sensível à história para *Shotgun Surgery* apresentada o seguinte raciocínio:

Shotgun Surgery: **SE** alterações na classe potencialmente afetam muita classes e muitos métodos e além disso é crescente o número de classes que herdam direta ou indiretamente dessa classe **ENTÃO POSSÍVEL Shotgun Surgery**

5.3.7

Divergent Change (DC)

Essa anomalia ocorre quando uma classe é modificada de diferentes formas por diferentes razões, ou seja, quando uma classe precisa ser modificada

para entrar em conformidade com alterações de diferentes naturezas. Ao explicar sobre tal problema, Fowler (1999) exemplifica que se você olha uma classe e observa que: terá que alterar três métodos todas as vezes que precisar capturar dados do banco e que terá que alterar outros quatro métodos todas as vezes que tiver que ajustar outros comportamentos dessa mesma classe, você estará diante de uma situação em que duas classes seriam melhor que uma.

De acordo com alguns princípios da orientação a objetos (Riel 1996) cada mudança em uma classe deve ser resultante de apenas um tipo de necessidade de alteração. Uma classe com *Divergent Change* quebra esse princípio. Se uma mesma classe é passível de ser alterada por diferentes motivos, é provável que ela seja uma classe mais instável e, portanto, mais propensa a erros. Dentre as estratégias apresentadas por Marinescu (Marinescu 2002, Marinescu 2004, Lanza e Marinescu 2006) não é apresentada nenhuma estratégia para a detecção dessa anomalia.

5.3.8 Estratégia de Detecção Sensível à História

Para a definição de uma estratégia sensível à história para a anomalia DC nos baseamos tanto em métricas convencionais quanto sensível à história. Entretanto, não seria possível nenhuma comparação direta com estratégias propostas pela literatura, pois até o momento nenhuma estratégia havia sido proposta para tal anomalia. A partir da definição apresentada para anomalia, optamos por avaliar a seguinte estratégia:

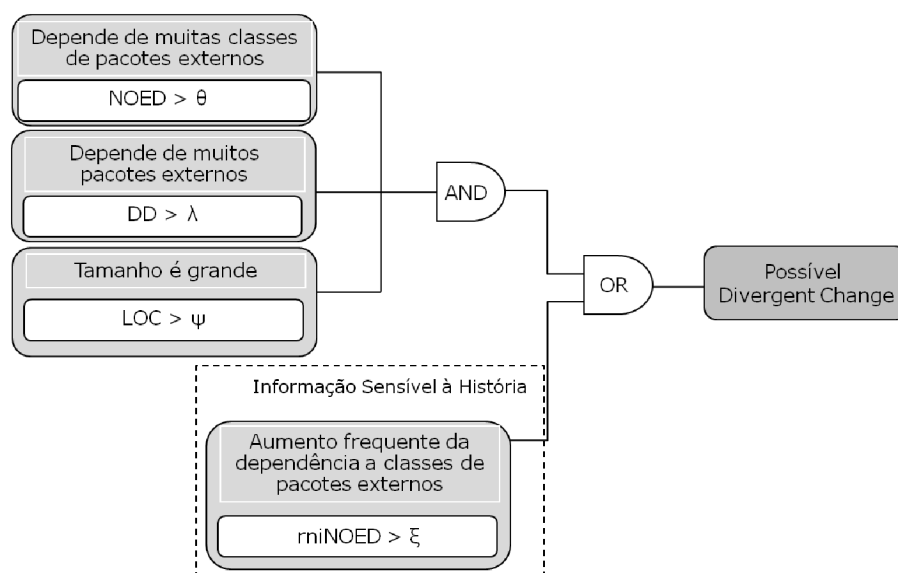


Figura 5.5: Estratégia de detecção sensível à história para *Divergent Change*

O raciocínio vinculado a tal estratégia está baseado na avaliação de propriedades como tamanho e dependência a classes de muitos pacotes externos.

A utilização da propriedade tamanho está baseada no fato de que, normalmente, classes grandes tendem a assumir muitas diferentes responsabilidades e, por isso, potencialmente sofreram alterações divergentes provenientes de cada uma dessas diferentes responsabilidades que são implementadas. Além disso, como pacotes tendem a agrupar módulos que possuem assuntos fortemente relacionados, a dependência a muitas classes pertencentes a muitos pacotes diferentes sugere que a classe avaliada aborda diferentes questões ao mesmo tempo. Tratam-se de hipóteses a serem verificadas na detecção dessa anomalia.

De forma sintetizada, a estratégia sensível à história para a anomalia DC apresentada o seguinte raciocínio:

Divergent Change: **SE** a classe depende de muitas outras classes de muitos distintos pacotes, é grande e, ao longo da história, a dependência a outros pacotes foi crescente **ENTÃO POSSÍVEL** *Divergent Change*