Table I
Defects detected in UML class diagrams (static)

No	Detected defects	Reaction	Severity
1	A generalization of an interface from a class was detected	Stop code generation	critical
2	A name of an element to be generated (e.g. a class, an operation, an attribute) is a keyword of C# language	Stop code generation	critical
3	A class relates via generalization to more than one general class	Stop code generation	critical
4	A cycle in class generalization was detected	Stop code generation	critical
5	A name of an element to be generated is missing	Generate the element pattern without its name. The element name has to be supplemented in the generated code	medium
6	A name of an element to be generated is not a valid C# name. It is assumed that white characters are so common shortcoming that they should be automatically substituted by an underline character	As above	medium
7	An interface visibility is *private* or *protected*	Use *package* visibility	low
8	A class visibility is *private* or *protected*.	Use *package* visibility	low
9	An interface is *abstract*	Treat the interface as no abstract	low
10	An interface has some attributes	Ignore attributes of the interface	low
11	An interface has nested classes	Ignore classes nested in the interface	low
12	A class that is no *abstract* has abstract operations	Treat the class as *abstract*	low

Table II.
Defects detected in UML state machines (static)

No	Detected defects	Reaction	Severity
1	A cycle in signal generalization was detected	Stop code generation	critical
2	A signal inherits after an element that is not another signal	Stop code generation	critical
3	A signal relates via generalization to more than one general signal	Stop code generation	critical
4	A region has more than one initial pseudostate	Stop code generation	critical
5	A state has more than one deep history pseudostate or shallow history pseudostate	Stop code generation	critical
6	There are transitions from pseudostates to the same pseudostates (different than a choice pseudostate)	Stop code generation	critical
7	There are improper transitions between orthogonal regions	Stop code generation	critical
8	A transition trigger refers to an nonexistent signal	Stop code generation	critical
9	An entry point, join or initial pseudostate has no incoming transition or more than one incoming transition	Stop code generation	critical
10	A deep or shallow history pseudostate has more than one outgoing transition	Stop code generation	critical
11	A transition from an entry/exit point to an entry/exit point	Stop code generation	critical
12	An exit point has no any incoming transition	Stop code generation	critical
13	Transitions outgoing a fork pseudostate do not target states in different regions of an orthogonal states	Stop code generation	critical
14	Transitions incoming to a join pseudostate do not originate in different regions of an orthogonal state	Stop code generation	critical
15	There is a transition originating in an initial pseudostate or a deep/shallow history pseudostate and outgoing a nested orthogonal state	Stop code generation	critical
16	The region at the topmost level (region of a state machine) has no initial pseudostate	Warn a user	medium
17	A transition outgoing a pseudostate has a trigger	Ignore the trigger	medium
18	A transition outgoing a pseudostate (different from a choice or junction vertex) has a nonempty guard condition	Ignore the guard condition	medium
19	A transition targeting a join pseudostate has a trigger or nonempty guard condition	Ignore the trigger and/or condition	medium
20	A trigger refers to a non-existing operation	The transition will be generated but it cannot be triggered by this event	medium
21	A trigger refers to an abstract operation or to an operation of an interface	as above	medium
22	A time event is deferred	Treat the event as not being deferred	medium
23	A final state has an outgoing transition	Warn a user	medium
24	A choice/junction pseudostate has at least one outgoing transition	Warn a user	medium

25	A choice/junction pseudostate has at least one incoming transition	Warn a user	medium
26	A join pseudostate has at least two incoming transitions	Warn a user	medium
27	A fork pseudostate has exactly one incoming transition and at least two outgoing transitions	Warn a user	medium
28	A terminate pseudostate has an outgoing transition	Warn a user	low

Table III.
Defects detected in UML state machines (dynamic)

No	**Detected defects**	**Reaction**	**Severity**
1	There is no enabled and no “else” transition outgoing a choice or junction pseudostate	Suspend execution - terminate	critical
2	A deep or shallow history pseudostate was entered that has no outgoing transitions and is “empty”, i.e. either a final state was a last active substate or the state was not visited before	Suspend execution - terminate	critical
3	More than one transition outgoing a choice or junction pseudostate is enabled	Select one enabled transition and ignore the others	medium
4	There is no enabled transition outgoing a choice or junction pseudostate and there is one or more “else” transition outgoing this pseudostate	Select one “else” transition and ignore other transitions	medium
5	More than one transition outgoing the same state is enabled	Select one transition and ignore the others	medium