

University of Pisa

MSc in Computer Engineering

Systems for Strategic Management and Support

LECTURE 9

<http://www.iet.unipi.it/m.cimino/pdis/>

Mario G.C.A. Cimino

Department of Information Engineering

Errors in BPMN models

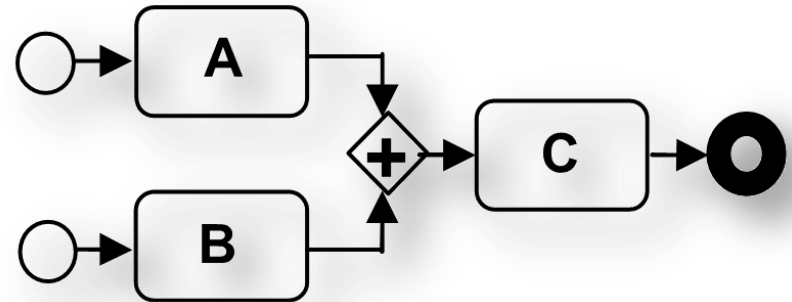
- ❑ **Syntactical errors:** to mistake the use of modeling elements. The valid combinations are usually prescribed by the standard. For instance, to use a control flow connecting two Pools.
- ❑ The syntactical correctness can be verified by using some modeling tools such as **Bonita**, **BizAgi**, **Intalio**, and so on.
- ❑ **Structural errors:** undesired run-time behavior, i.e., the model does not fulfill important structural criteria. Deadlock, livelock, and multiple terminations are the most important, as can produce expensive damages.
- ❑ **Deadlock:** a deadlock in a process model is given if a certain instance of this model cannot continue working, while it has not reached the process end.
- ❑ **Livelock:** Livelock is as a state from which it is possible to proceed, but it may be impossible to reach the desired final state.
- ❑ **Multiple termination:** is a situation where an instance of the process cannot terminate in one of the predefined end events.

Errors in BPMN models: examples

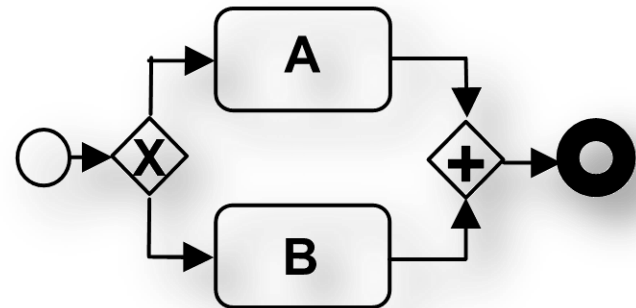
❑ **Loop deadlock:** there is an execution path from the output of an AND-join back to its input points.



❑ **Multi source deadlock:** two independent sources lead at the input points of AND-join gateway.

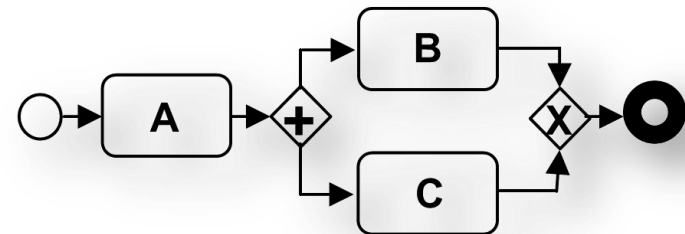
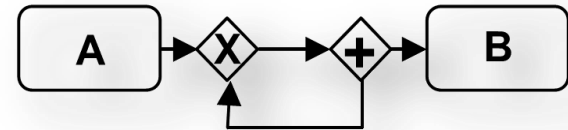


❑ **Improper structuring deadlock:** an AND-join gateway receives input that early started from an XOR-split.



Errors in BPMN models: examples

- ❑ **Livelock:** an infinite execution of process, happening when an AND-split is used instead of an XOR-split for loop modeling.
- ❑ **Multiple terminations:** a situation where exists an AND-split before an XOR-join gateway.



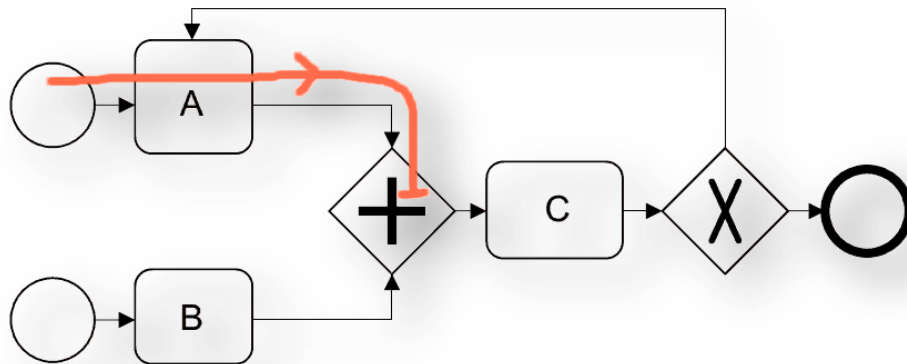
- ❑ **Message-related mismatch**

- a) *message number mismatch*: the number of messages provided differs from the number of messages expected
- b) *message type mismatch*: the message type provided differs from the message type expected
- c) *message format mismatch*: the message format provided differs from the message format expected

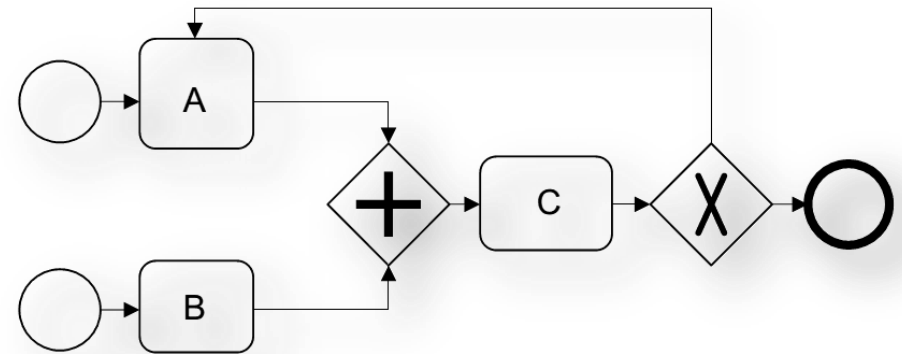
Errors in BPMN models: examples

- ❑ A **counterexample** is an exception to a certain property to demonstrate that it does not hold for all process instances.

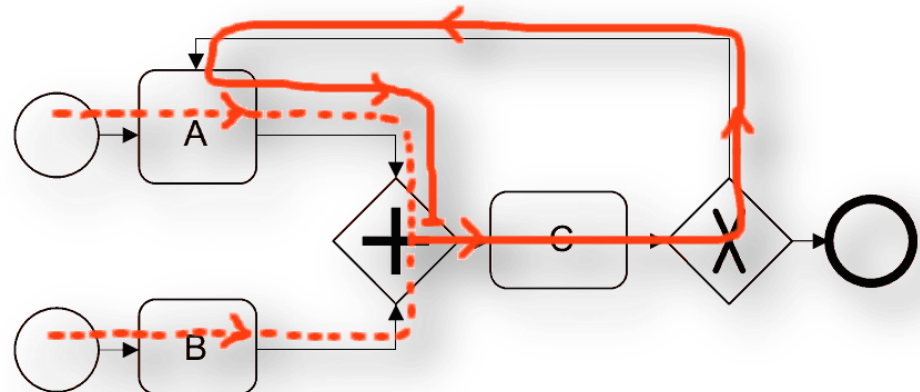
- ❑ Is the following process model deadlock-free? →



- ❑ Second counterexample (improper structuring) →

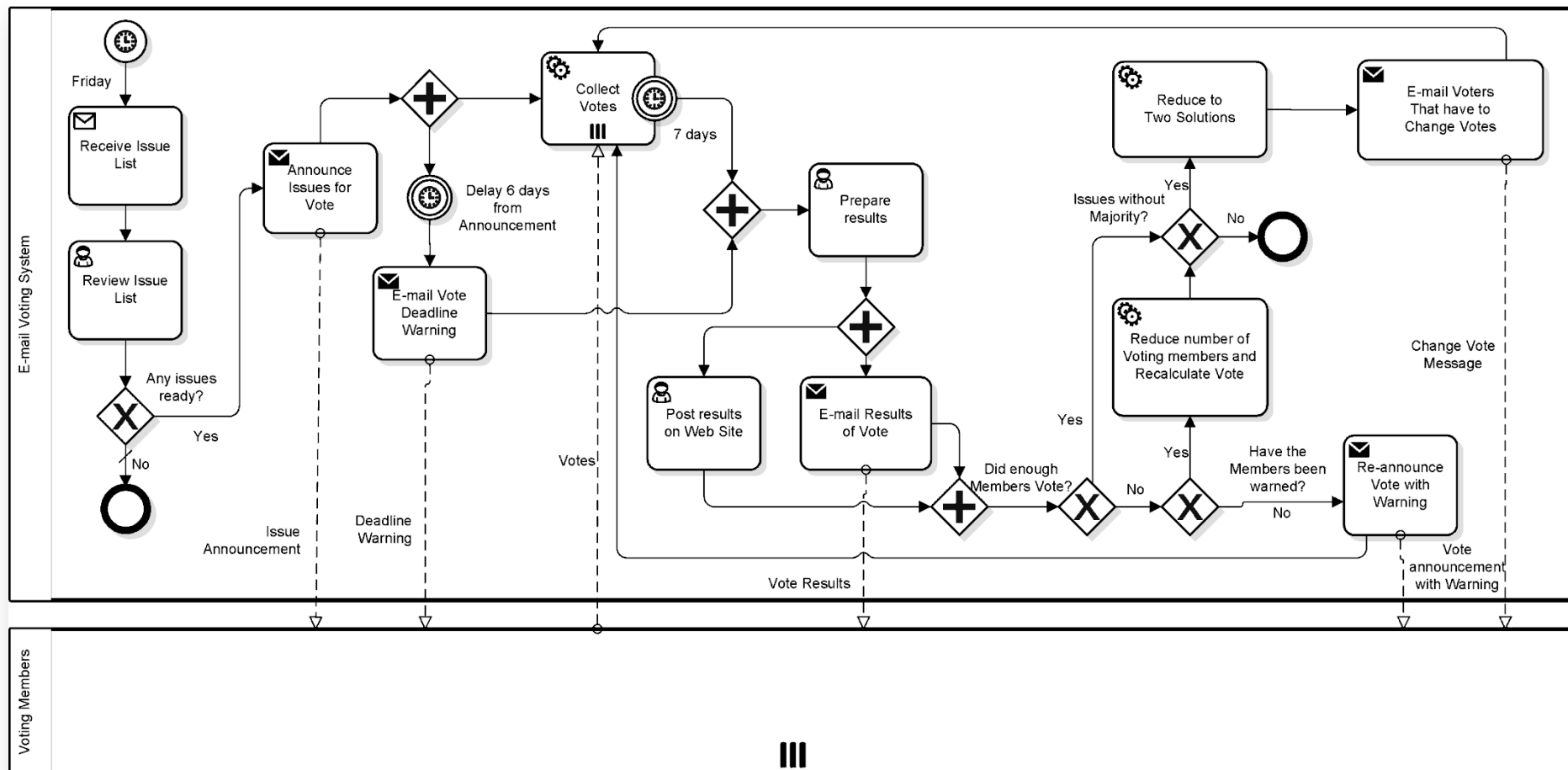


First counterexample ❑
← (multi source deadlock)



E-mail voting system

- ❑ Describe in semi-formal natural language the following BPMN process, representing the management of an E-mail Voting system.
- ❑ Define what token represents.
- ❑ Is it deadlock-free?



E-mail voting system

□ Given 100 starting tokens, determine the number of ending tokens for each scenario (path), considering the following branching proportions at each gateway: no issues ready (10%), enough members voted (96%), issues with majority (4%).

1. It is Friday

2. The System receives the Issue List (receive task)

3. The System reviews the Issue List (user task)

4.1 If there are no issues ready for vote:

4.1.1 End

4.2 If there are issues ready for vote:

4.2.1 The System announces issues for vote (send task)

4.2.2 The System sends the issue announcement to Voting Members
(The Voting Members receive the announcement from the System)

4.2.3.a.1 Six days have been elapsed since the announcement

4.2.3.a.2 The System sends to Members a deadline warning for vote (send task)
(The Members receive the deadline warning)

4.2.3.a.3 → 4.2.4

E-mail voting system

...

4.2.3.b.1 (The Voting Members send votes to the System)

The System collects votes (parallel service task)

4.2.3.b.2 7 days have been elapsed since the start of the vote collection

4.2.4 The System waits for the end of both flows 4.2.3 a and b

4.2.5 The System prepares results (user task)

4.2.6.a The System posts results on Web Site (user task) → 4.2.7

4.2.6.b The System sends results of vote to Voting Members (send task)

(Voting Members receive results of vote from the System)

4.2.7 The System waits for the end of both flows 4.2.6 a and b

4.2.8.1 If enough members vote → 4.2.9

4.2.8.2 If not enough members vote

4.2.8.2.1 If members have been warned

4.2.8.2.1.1 Reduce the number of voting members and recalculate vote
(service task) → 4.2.9

...

E-mail voting system

...

4.2.8.2.2 If members have not been warned

4.2.8.2.2.1 The System re-announce vote with warning to Voting Members (send t.)
(The Voting Members receive the vote with warning from the System)

4.2.8.2.2.2 → 4.2.3.b.1

4.2.9 (go on)

4.2.9.1 If there are no issues with majority → End

4.2.9.2 If there are issues with majority

4.2.9.2.1 The System reduces to two solutions (service task)

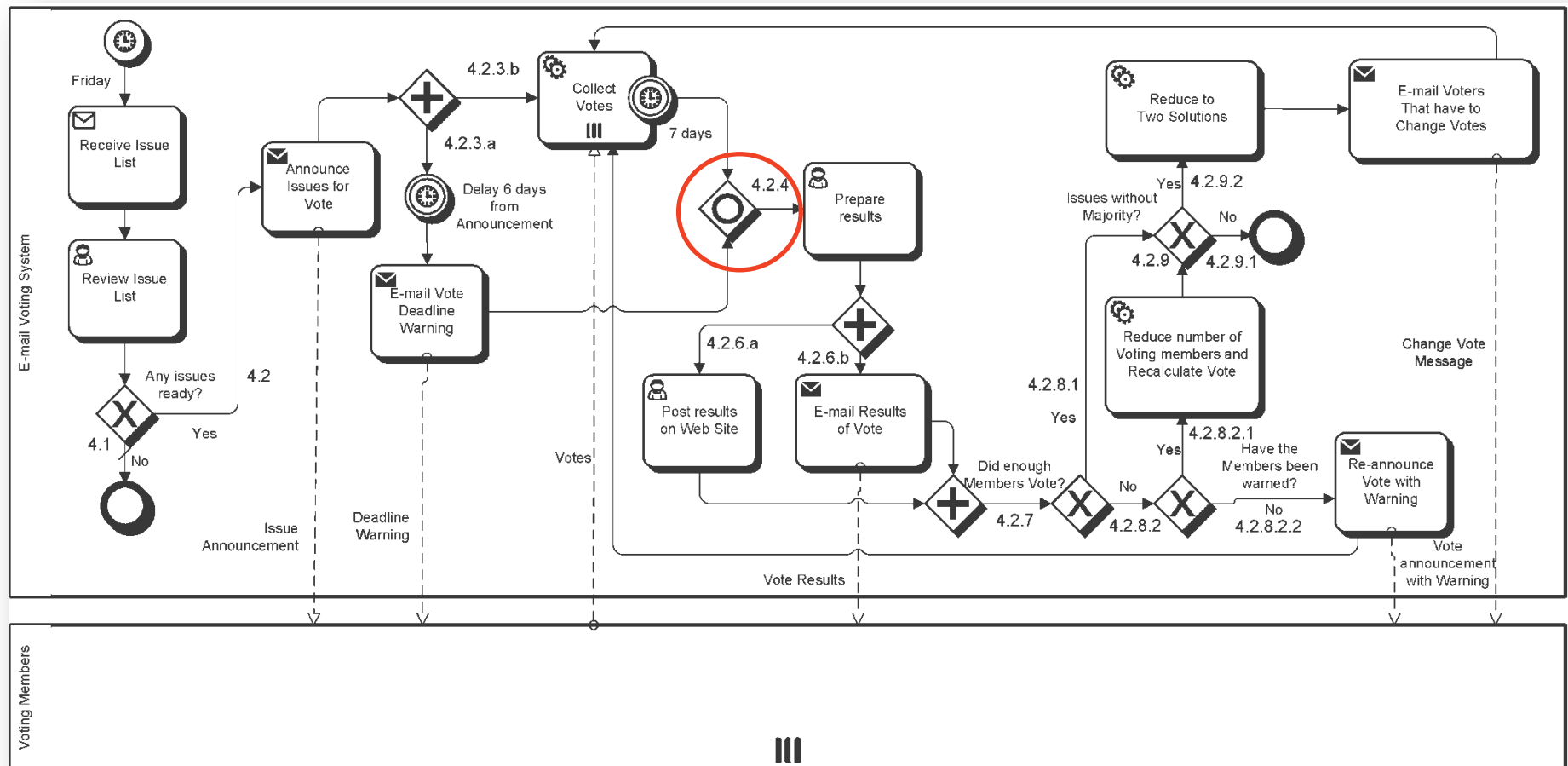
4.2.9.2.2 The System emails only to Voting Members that have to change votes
(send task)
(Voting Members receive the change vote message from the System)

4.2.9.2.3 → 4.2.3.b.1

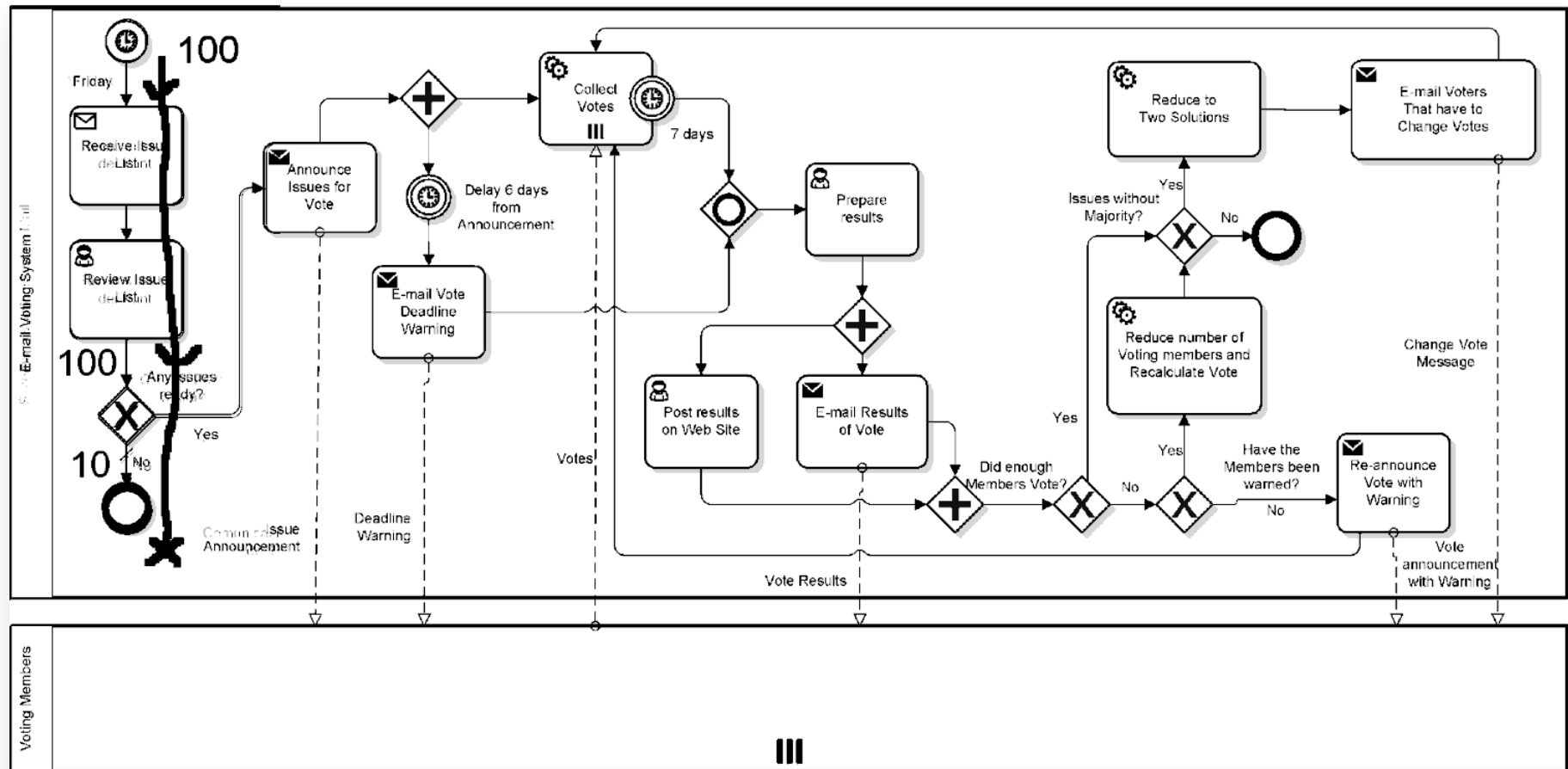
- A token represents an “issue list”. Indeed, all tasks can be supposed to be carried out on an issue list: each voting member votes all issues of the list when voting (e.g. via a web form); results are also related with an issue list; majority is intended for all issues; the issues without majority make a new issue (sub-)list, with the two most used solutions for each issue.

E-mail voting system

- ❑ The structural error is represented by the parallel join gateway 4.2.4: it should be replaced by a inclusive join gateway. It causes a deadlock when the flow comes back from either “Email Voters that have to change votes” or “Re-announce vote with warning”. Indeed for both cases the branch 4.2.3.a is not working, and then the parallel gateway 4.2.2 is expected to wait indefinitely.

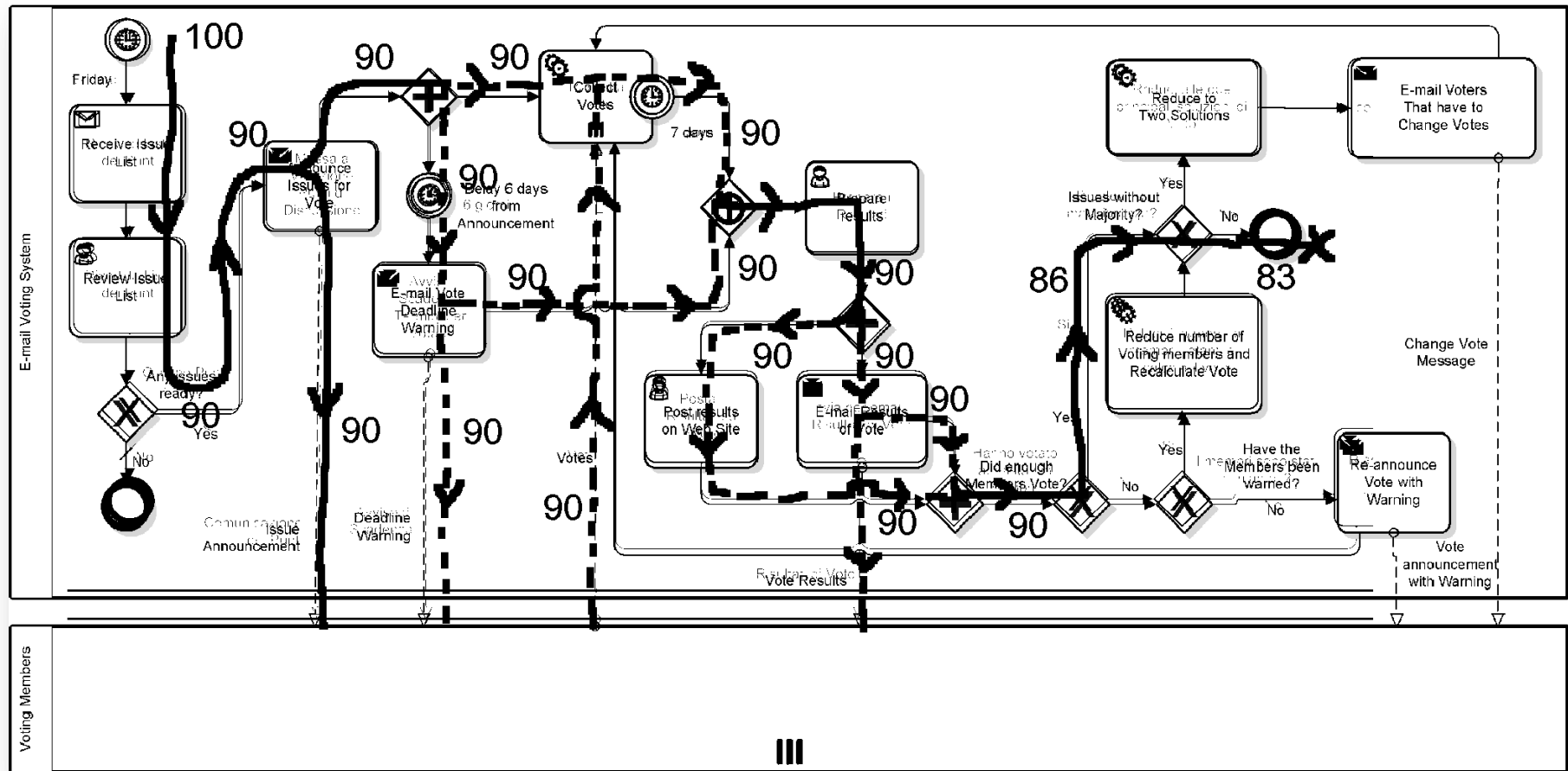


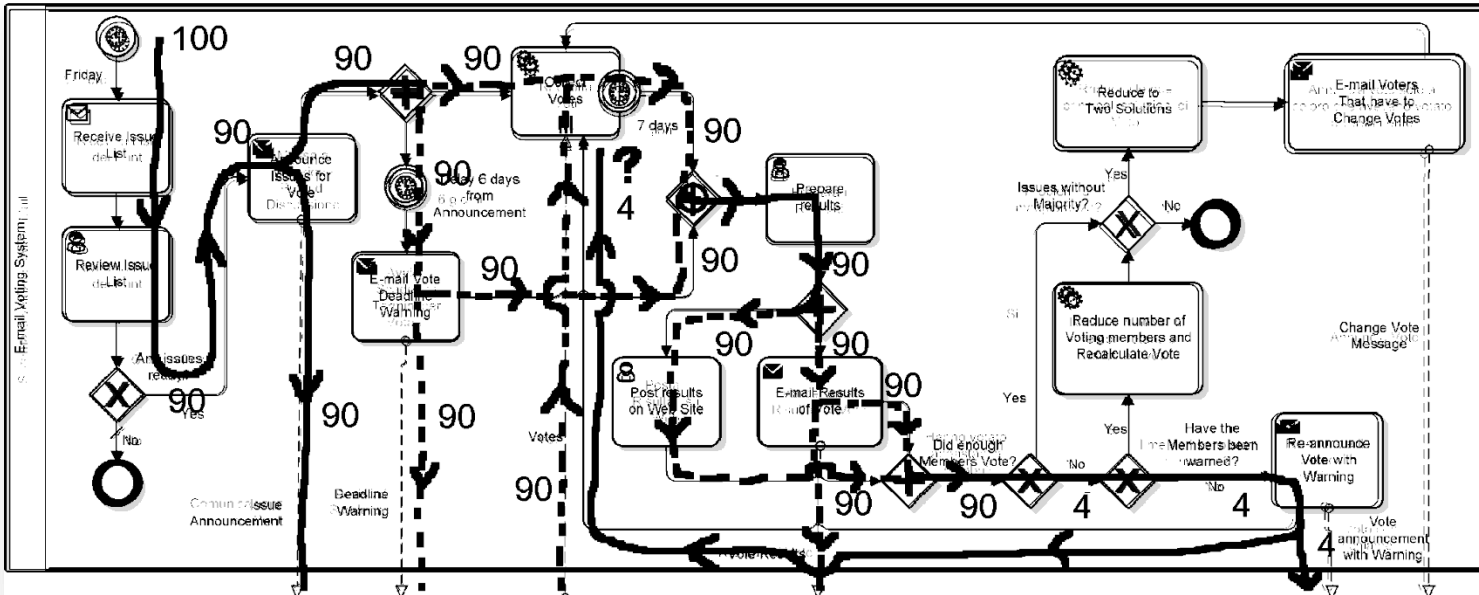
S1) NO ISS. READY: $100 \times 0.1 = 10$.



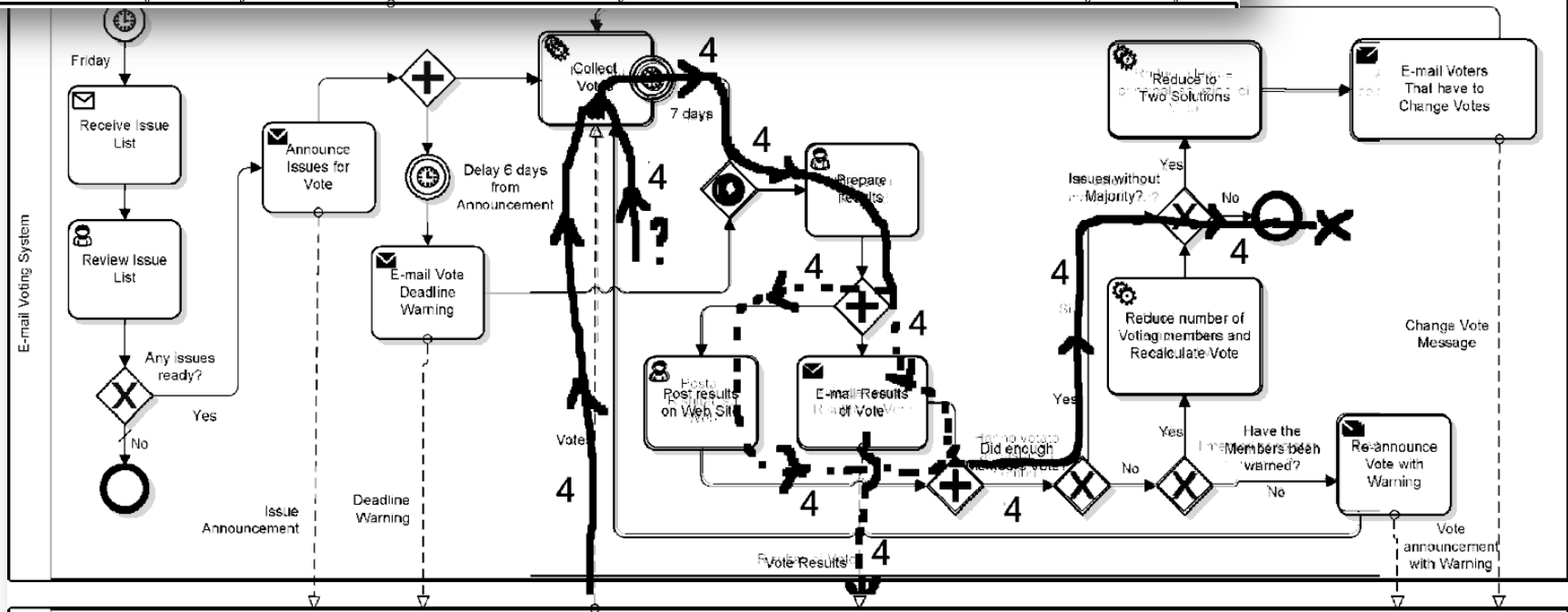
S2) ISS. READY & ENOUGH M. & NO MAJ. ISS.:

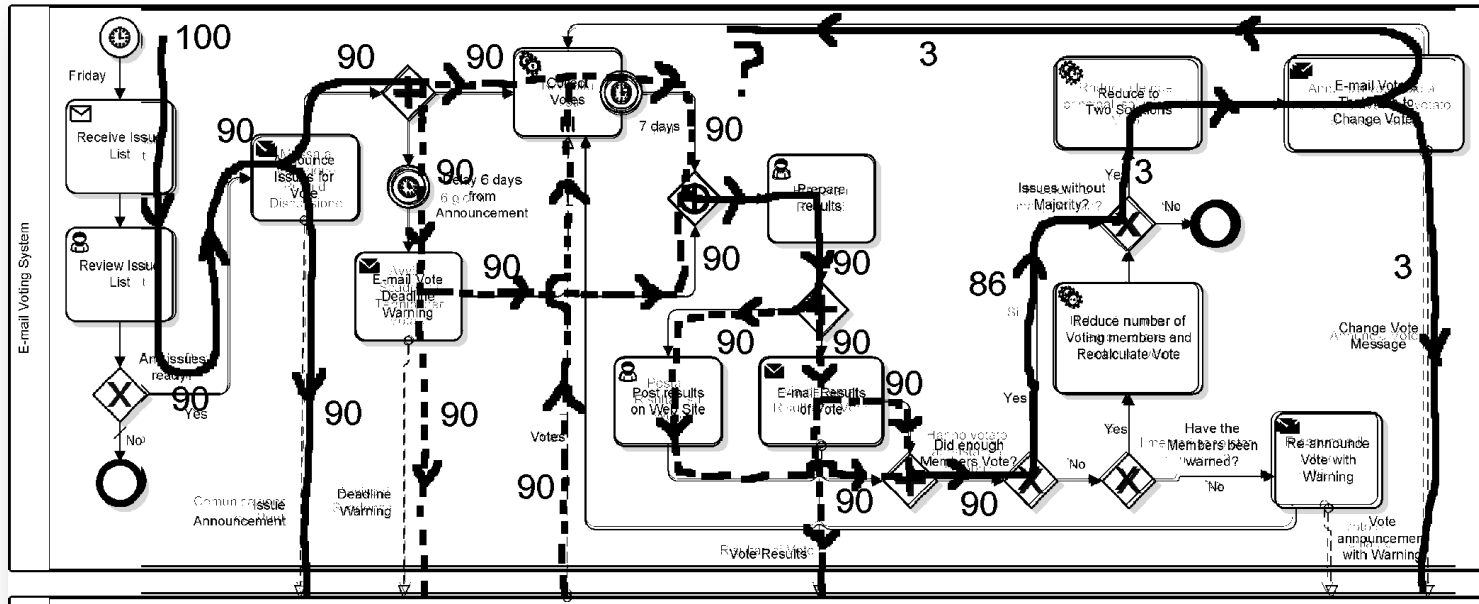
$$100 \times .9 \text{ \& } 90 \times .96 \text{ \& } 86 \times .96 = \mathbf{83}.$$





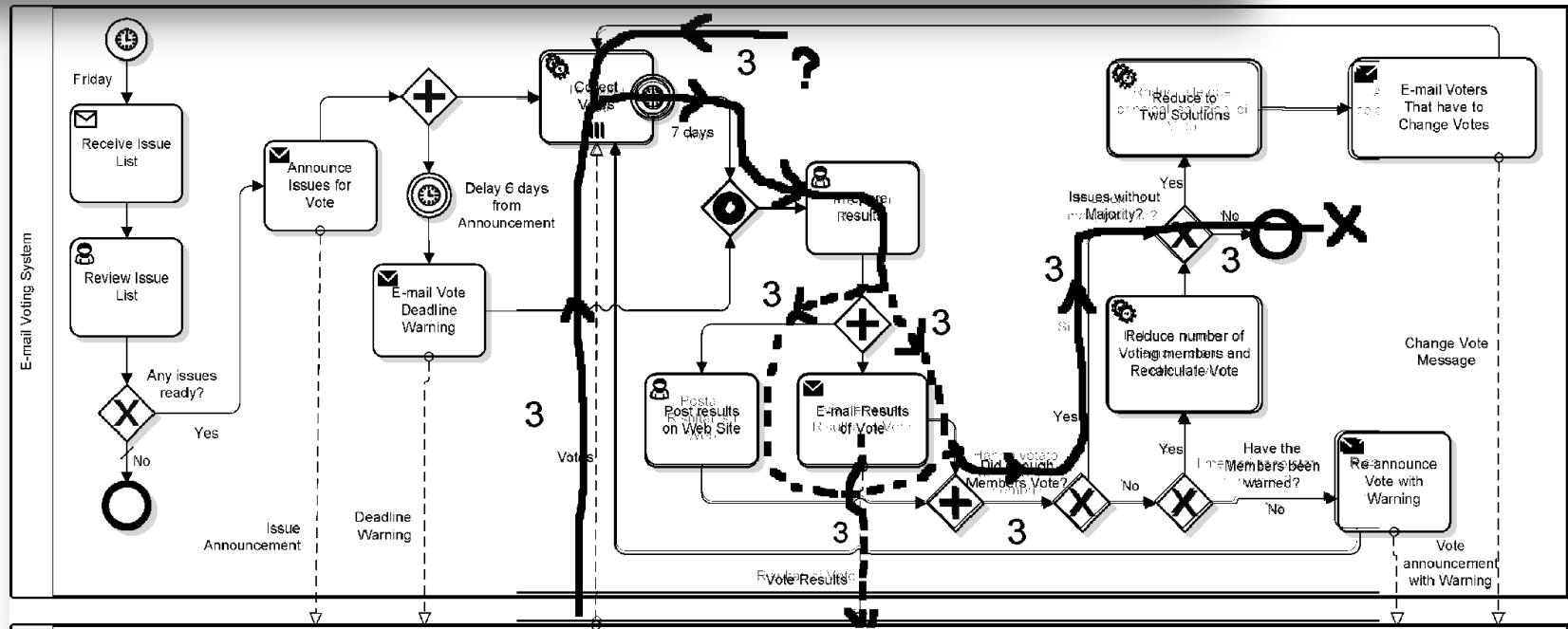
S3) ISS. READY &
NO ENOUGH M. &
NO WARN &
ENOUGH M. &
NO MAJ. ISS.:
 $100 \times .9$ &
 $90 \times .04$ &
 4 &
 4 &
 $4 \times .96$
 $= 4.$





S4) ISS. READY & ENOUGH M. & MAJ. ISS. & ENOUGH M. & NO MAJ. ISS.:
 $100 \times .9$ &
 $90 \times .96$ &
 $86 \times .04$ &
 $3 \times .96$ &
 $3 \times .96$
 = 3.

$$\begin{aligned} S1 + S2 \\ + S3 + S4 \\ = 10 + 83 \\ + 4 + 3 \\ = 100. \end{aligned}$$



Questions

1. Votes can be re-announced with warning many times.
☐ true ☐ false
2. The process can be completed even if the most people do not vote.
☐ true ☐ false
3. The process can be completed even if voting members disagree.
☐ true ☐ false

Answers

1. **False:** re-announcement with warning cannot be carried out any more after being performed the first time, because of the exclusive gateway “have the members been warned?”
2. **True:** when the most people do not vote, the question “did enough members vote” is false, and then the system continues with “reduce number of voting members and recalculate vote”, so as to end the process even if only two people voted.
3. **True:** if major solution does not emerge in the first “ballot”, members who voted for minor solutions will have to choose among the two major solutions.

In the example below, a new reduced issue list will be sent to some members, each issue being constrained in terms of answers for each member:

	Issue 1: x y?	Issue 2: w	Issue 3: h j?	Issue 4: r
Member A	answer x	answer w	answer h	answer r
Member B	answer y	answer q	answer j	answer r
<i>Member C</i>	<i>answer z</i>	answer w	answer h	answer t
Member D	answer x	answer w	answer j	answer r
<i>Member E</i>	answer y	answer t	<i>answer k</i>	answer s

- Member C is asked to answer again issue 1, choosing x or y
- Member E is asked to answer again issue 3, choosing h or j